

北京交通大學 国家级精品课程

信号与系统

Signals and Systems

电子信息工程学院

信号与系统教研室

2008年

利用MATLAB进行信号与系统分析

MATLAB简介

(Matrix Laboratory)

MATLAB的工作方式

如何获取帮助

表达式——变量、数值、算数运算符、关系运算符、逻辑运算符、冒号运算符

数组及其运算

函数文件

循环（**FOR**、 **WHILE** 循环）

基本绘图语句

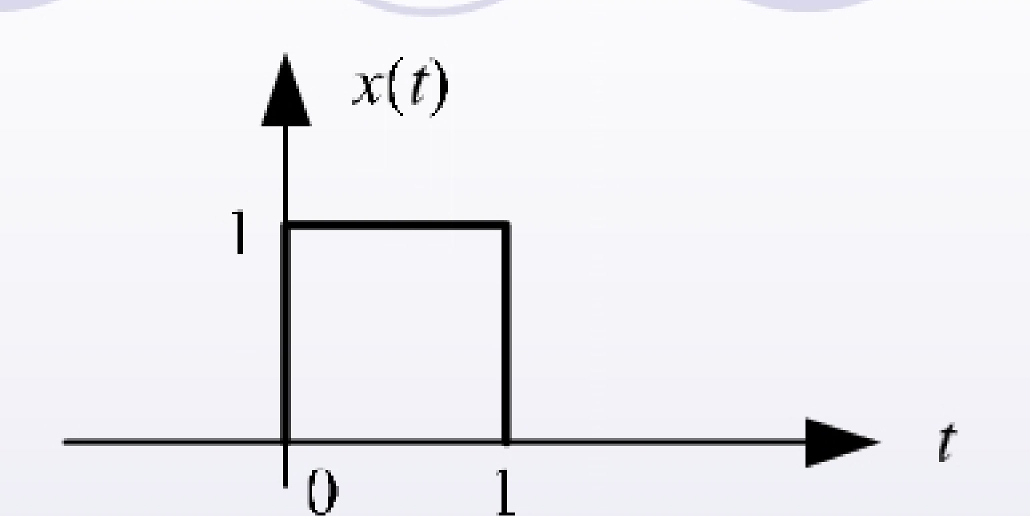
一、MATLAB的工作方式

（1）窗口命令方式

（2）运行以 .M 为扩展名磁盘文件

工作方式举例

```
%用plot函数画一个方波
t=[−1 0 0 1 1 3];
x=[0 0 1 1 0 0];
plot(t,x);
xlabel('t');ylabel('x(t)');
axis([−1 3 0 2]);
```



直接在命令窗口输入以上命令

建一个名字为my_file.M的文件，然后在命令窗口输入文件名及回车。

二、获取帮助

命令窗口输入：　help+函数名

例如　help plot

三、表达式

- 变量
  - ✓ 不需要变量的类型说明
  - ✓ 变量名的第一个字符必须是字母
  - ✓ 变量名长度：不超过31个字符
  - ✓ 大写和小写的字母视为不同的字符

    例如：num_students = 25
  - ✓ 特殊变量：

    pi 表示圆周率，inf 表示无穷大，NaN(Not a Number)表示不定量，如0/0。

三、表达式

- 数值
  - ✓ MATLAB用常规的十进制表示数值
  - ✓ 用i或j作为后缀来表示复数的虚部

例 1.235e5表示1.235×10^5，x=2+3j

abs(x)	求复数x的模
angle(x)	求复数x的相角(弧度)
real(x)	求复数x的实部
imag(x)	求复数x的虚部
conj(x)	求复数x的共轭

三、表达式

- 运算符号
  - 算数运算符
    - \+ 加
    - − 减
    - * 乘
    - / 除
    - ^ 乘方
    - ‘ 矩阵的复共轭转置

三、表达式

- 运算符号
  - 逻辑运算符
    - A & B　　逻辑与(and)
    - A | B　　逻辑或(or)
    - ~A　　逻辑非(not)

值为0时表示逻辑假(F)，其它任何非零值表示逻辑真。

三、表达式

- 运算符号
  - 关系运算符
    - A < B　　小于
    - A > B　　大于
    - A <= B　　小于等于
    - A >= B　　大于等于
    - A == B　　等于
    - A ~= B　　不等于

三、表达式

- 运算符号
  - 冒号运算符

◆ 表达式 1:10 表示产生一个行向量，它的值为

1 2 3 4 5 6 7 8 9 10

◆ 表达式 10:–2:1 表示产生一个递减的行向量，它的值为

10 8 6 4 2

四、数组

1. 数组的构造

用冒号:产生数组

例 x=2:5 产生一个数组，它的值为

x(1)=2, x(2)=3, x(3)=4, x(4)=5

用linspace产生数组

x=linspace(0,2,11)将区间[0，2]均匀抽样11点作为数组x

给2维数组赋值时，用分号表示一行的结束，如：z=[1 2; 3 4]。

四、数组

1. 数组的构造

MATLAB 提供了一些产生基本矩阵的函数

zeros	产生矩阵元素全为0的矩阵
ones	产生矩阵元素全为1的矩阵
rand	产生(0,1)均匀分布随机数矩阵
randn	产生正态分布随机数矩阵

四、数组

2. 数组的运算

- ✓ 数组和一个标量相加或相乘

  例　y=x−1　z=3*x
- ✓ 2个数组的对应元素相乘除 .* ./

  例　z=x.*y
- ✓ 确定数组大小的函数

  size(A)　返回值数组A的行数和列数（二维）

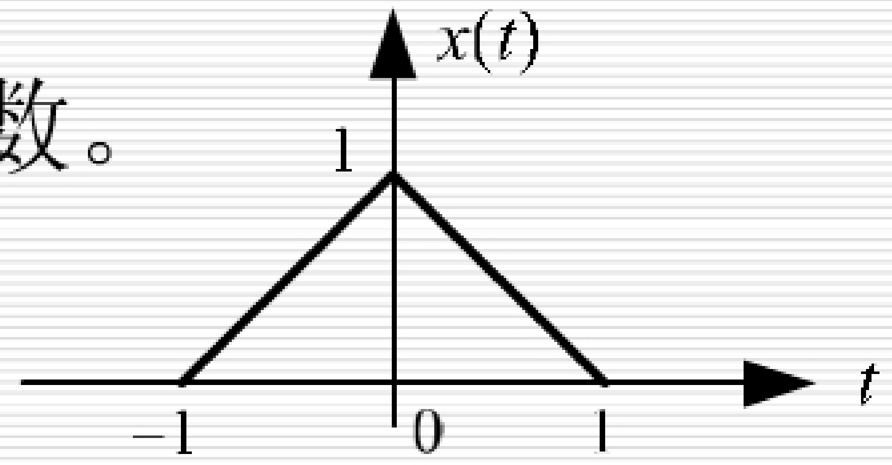
  length(B)　确定数组B的元素个数（一维）

五、函数文件

- M文件的第一行包含**function**
- 功能：建立一个函数，可以同MATLAB的库函数一样使用。

五、函数文件

例：编一个绘制图示波形的函数。




```
function y=tri(t)
y=[ abs(t)<=1].*(1−abs(t));
```

调用函数tri，并画出它的波形

```
t=−2:0.05:2;
plot(t,tri(t));
```

六、For 循环

例：编写计算s=1+2+3+…+100的MATLAB程序

```
s=0;
for n=1:100
   s=s+n;
end
```

七、While 循环

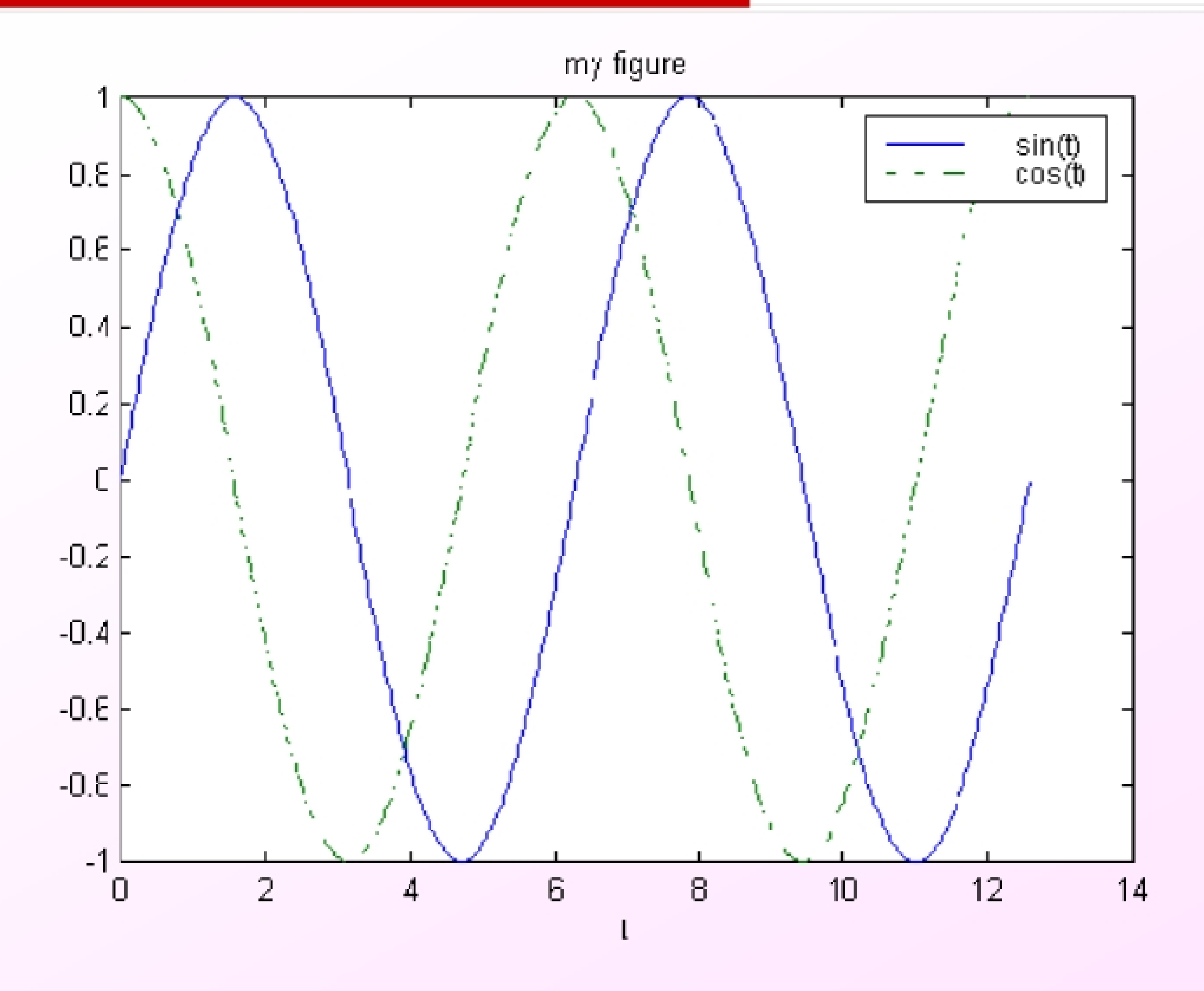
例：计算 $s=\frac{1}{1^2}+\frac{1}{2^2}+\frac{1}{3^2}+\cdots$ 的值，且误差小于10^{-6}

```
s=0;
n=1;
eps=1e–6;
while 1/(n*n) > eps
  s=s+1/(n*n);
  n=n+1;
end
fprintf('s=%.5f\n',s)
```

八、plot函数——绘图函数(continuous)

```
t=linspace(0,4*pi,512);
plot(t,sin(t),t,cos(t),'-.');
title('my figure');
xlabel('t');
legend('sin(t)','cos(t)');
```

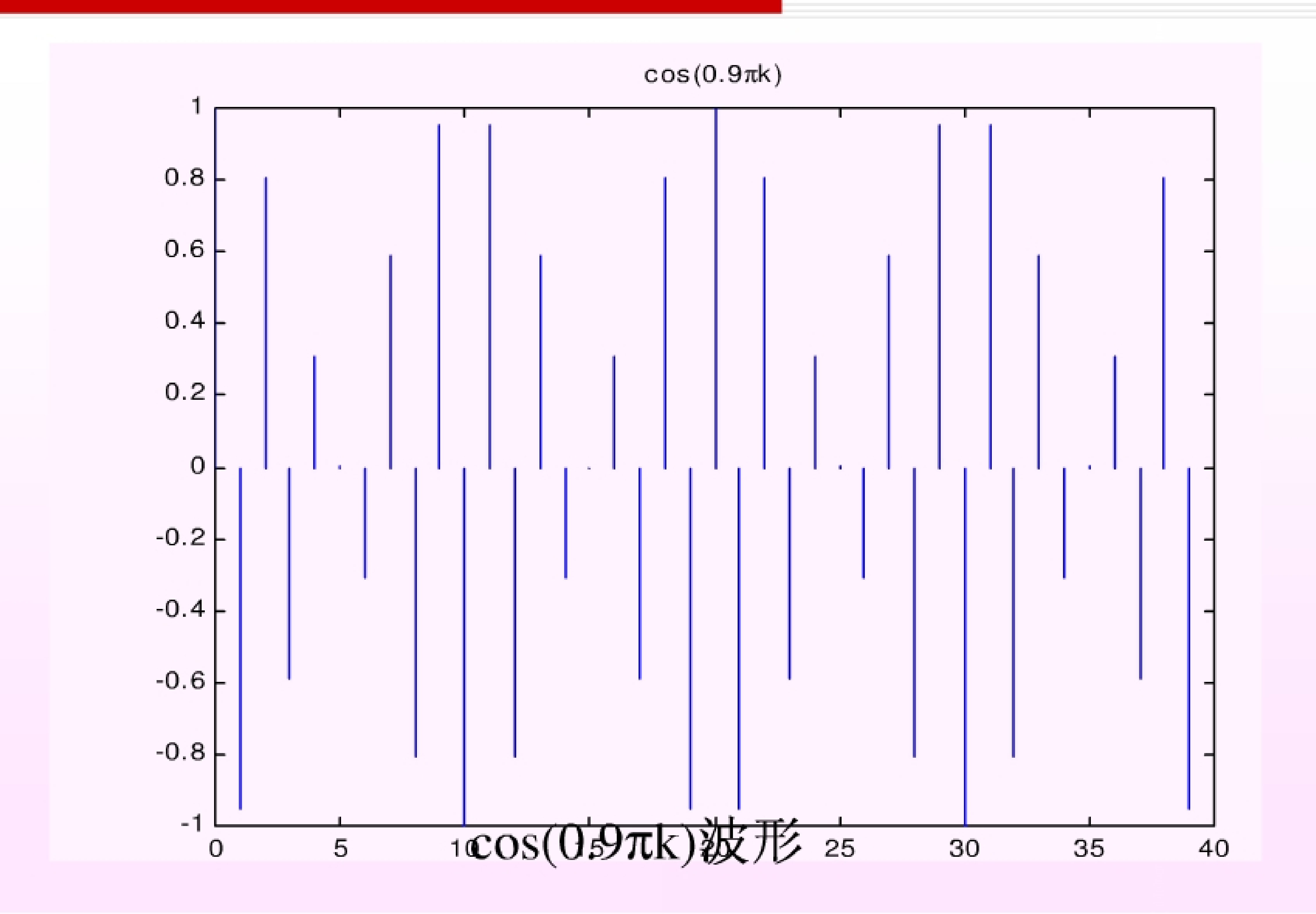
八、plot函数——绘图函数(continuous)



九、stem函数——绘图函数(discrete)

```
k=0:39;
stem(k,cos(0.9*pi*k));
title('cos(0.9\pik)');
```

九、stem函数——绘图函数(discrete)



cos(0.9πk)波形



信号的MATLAB表示

- 基本信号的**MATLAB**表示

  指数信号Ae^{at}、指数序列a^k、抽样函数Sa(t)、正弦型信号、矩形脉冲信号、三角脉冲信号

- 信号基本运算的**MATLAB**实现

  尺度变换、翻转、时移、 相加、相乘、差分与求和、微分与积分

一、基本信号的MATLAB表示

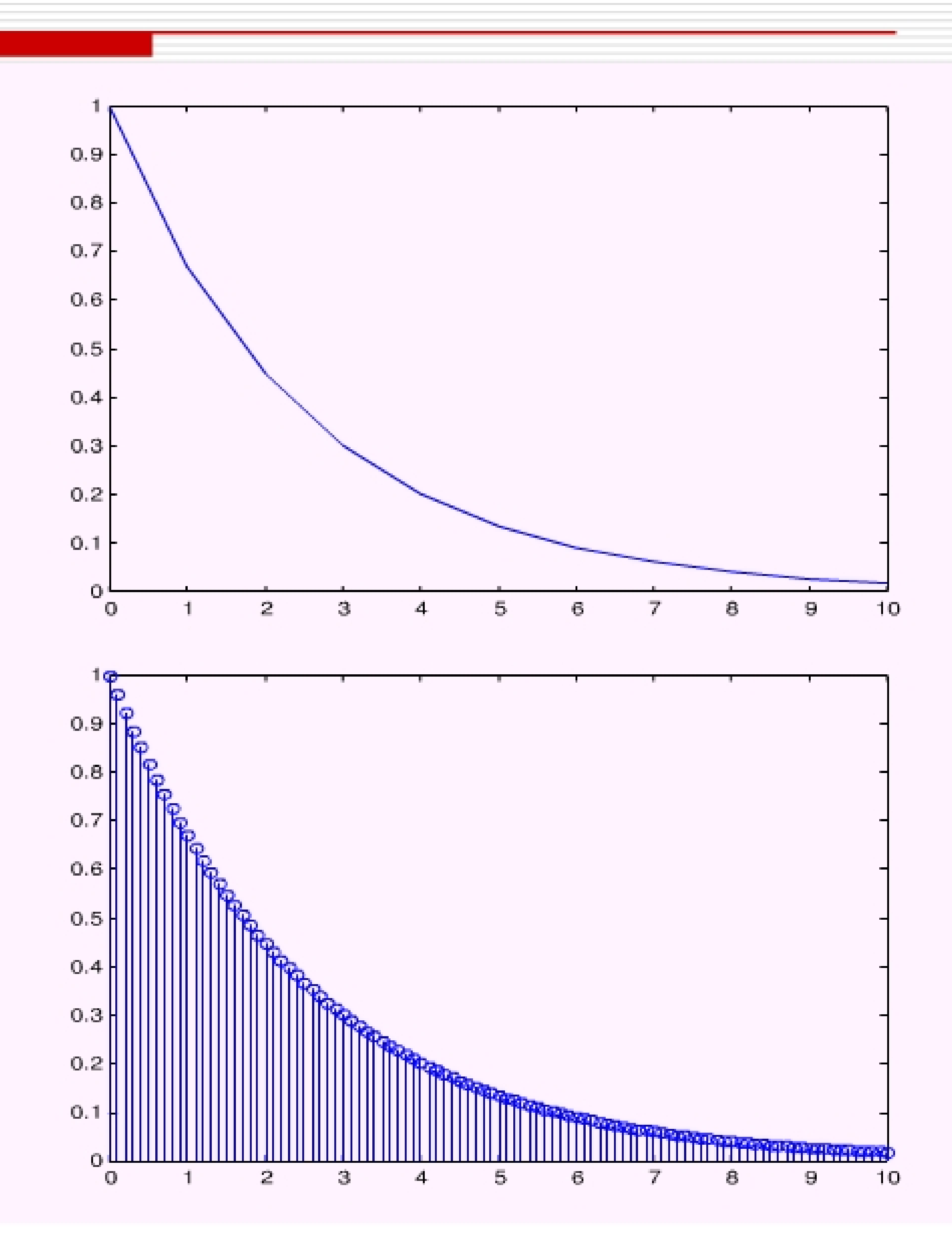
- 指数信号Ae^{at}　　y = A*exp(a*t);
- 指数序列a^k　　幂运算a.^k实现
- 正弦型信号　　内部函数cos( ) 和sin( )
- 抽样函数Sa(t)　　sinc(t)
- 矩形脉冲信号　　y = rectpuls(t,width)
- 三角波脉冲信号

  y = tripuls(t, width,skew)

一、基本信号的MATLAB表示

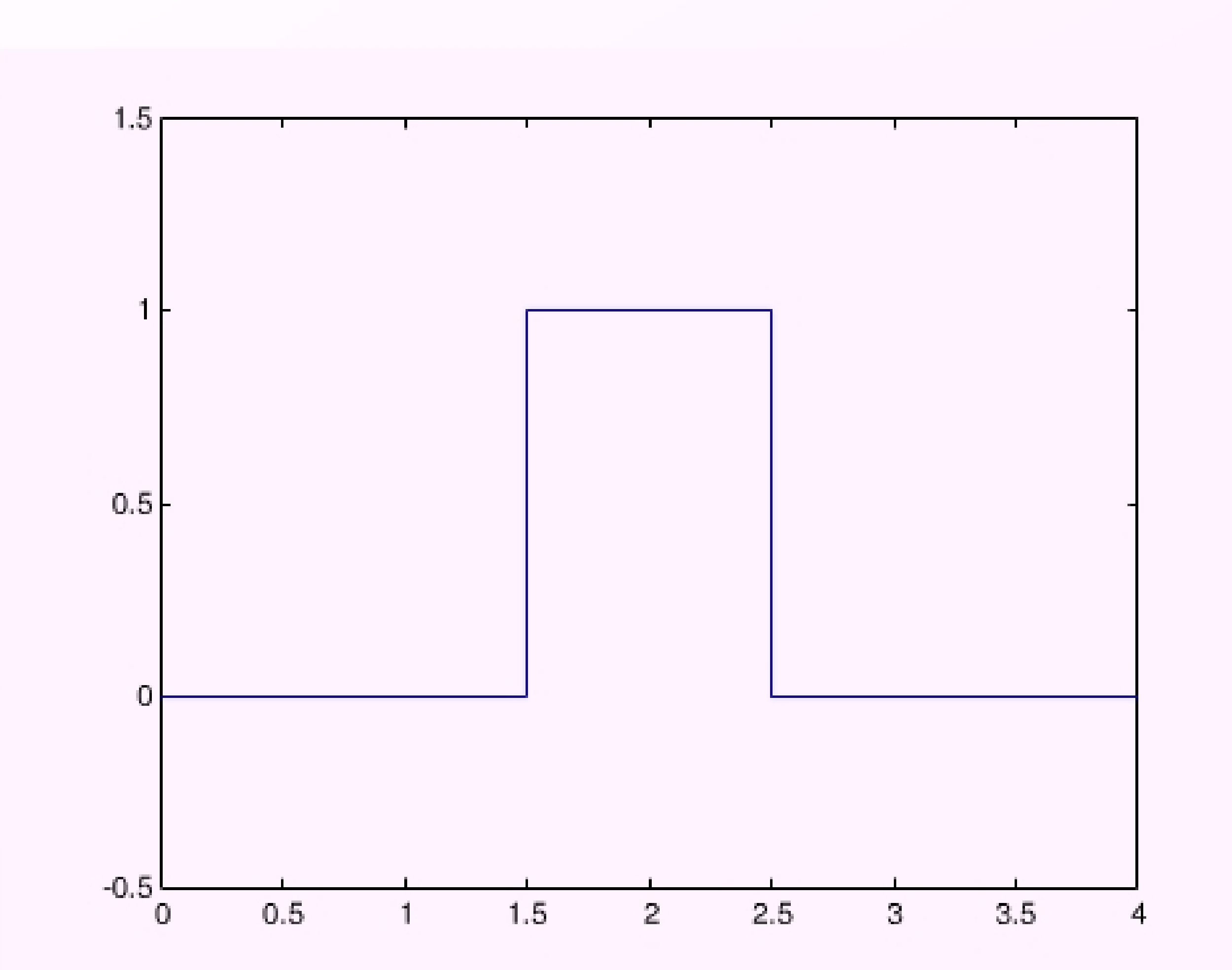
```
%decaying exponential
 t=0:001:10;
 A=1;
 a=-0.4;
 ft=A*exp(a*t);
 plot(t,ft)

 t=0:0.1:10;
 A=1;
 a=-0.4;
 ft=A*exp(a*t);
 stem(t,ft)
```



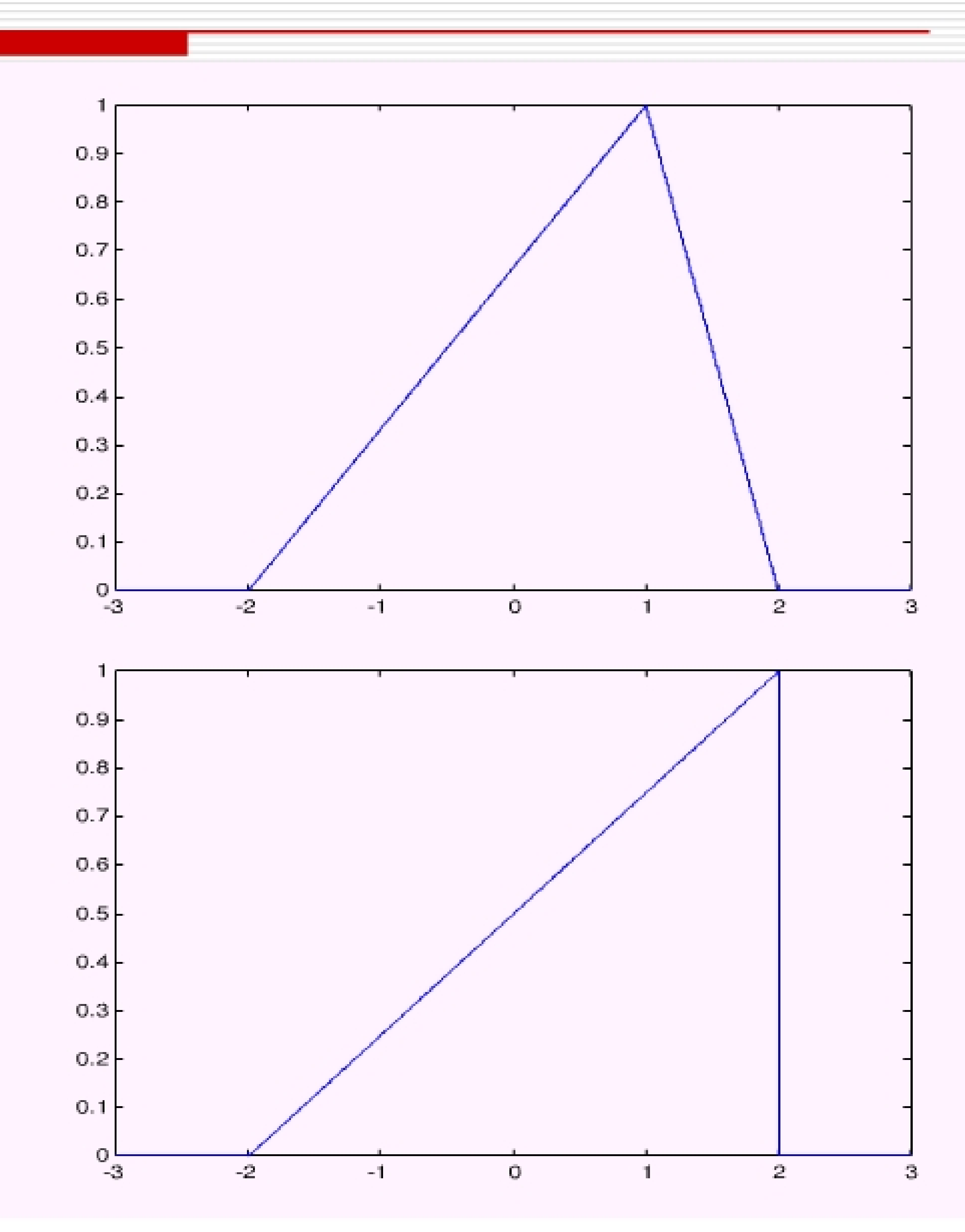
一、基本信号的MATLAB表示

```
% rectpuls
t=0:0.001:4;
T=1;
ft=rectpuls(t-2*T,T);
plot(t,ft)
axis([0,4,-0.5,1.5])
```



一、基本信号的MATLAB表示

```
% tripuls
t=-3:0.001:3;
ft=tripuls(t,4,0.5);
plot(t,ft)
```




```
ft=tripuls(t,4,1);
```

一、基本信号的MATLAB表示

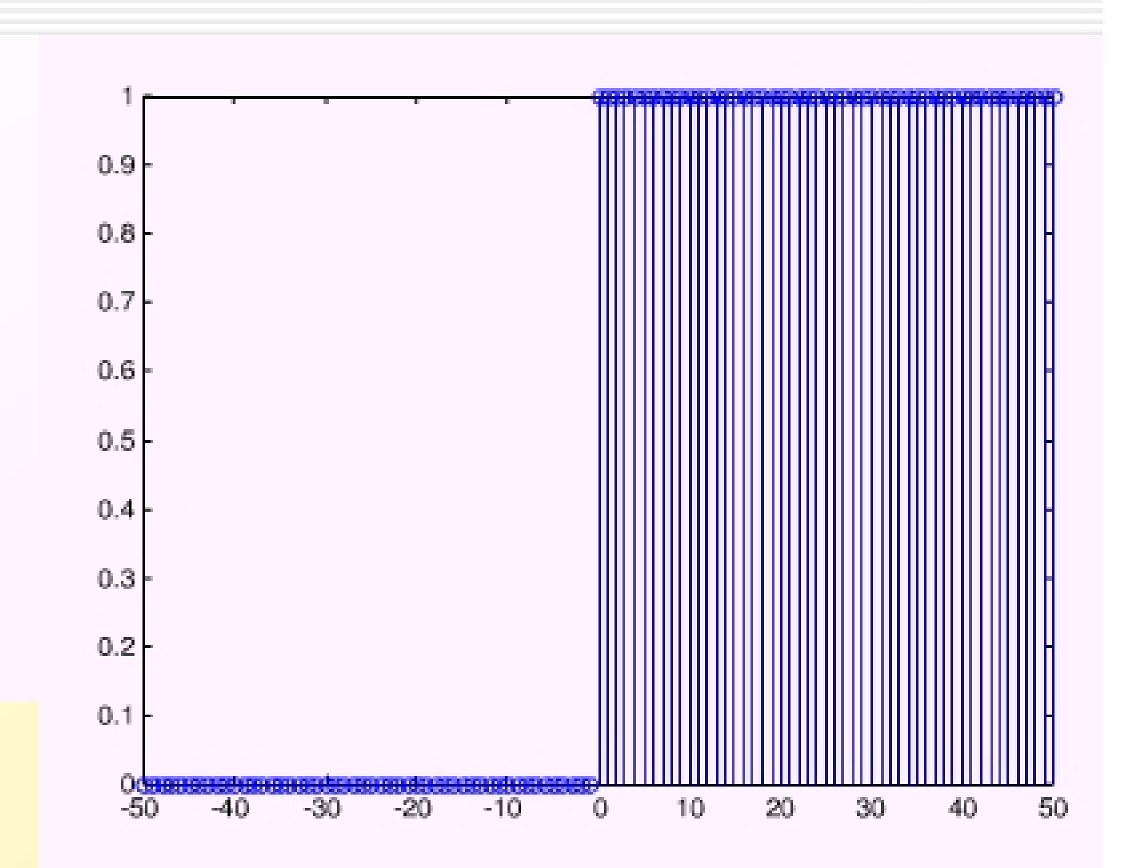
```
% unit impuls sequence
k=-50:50;
delta=[zeros(1,50),1,zeros(1,50)];
stem(k,delta)
```

```
function [f,k]=impseq(k0,k1,k2)
%产生 f[k]=delta(k−k0);  k1<=k<=k2
k=[k1:k2];f=[(k−k0)==0];
```

```
k0=0;k1=−50;k2=50;
[f,k]=impseq(k0,k1,k2);
stem(k,f)
```

一、基本信号的MATLAB表示

```
% unit step sequence
k=-50:50;
uk=[zeros(1,50), ones(1,51)];
stem(k,uk)
```




```
function [f,k]=stepseq(k0,k1,k2)
%产生 f[k]=u(k-k0);k1<=k<=k2
k=[k1:k2];f=[(k-k0)>=0];
```

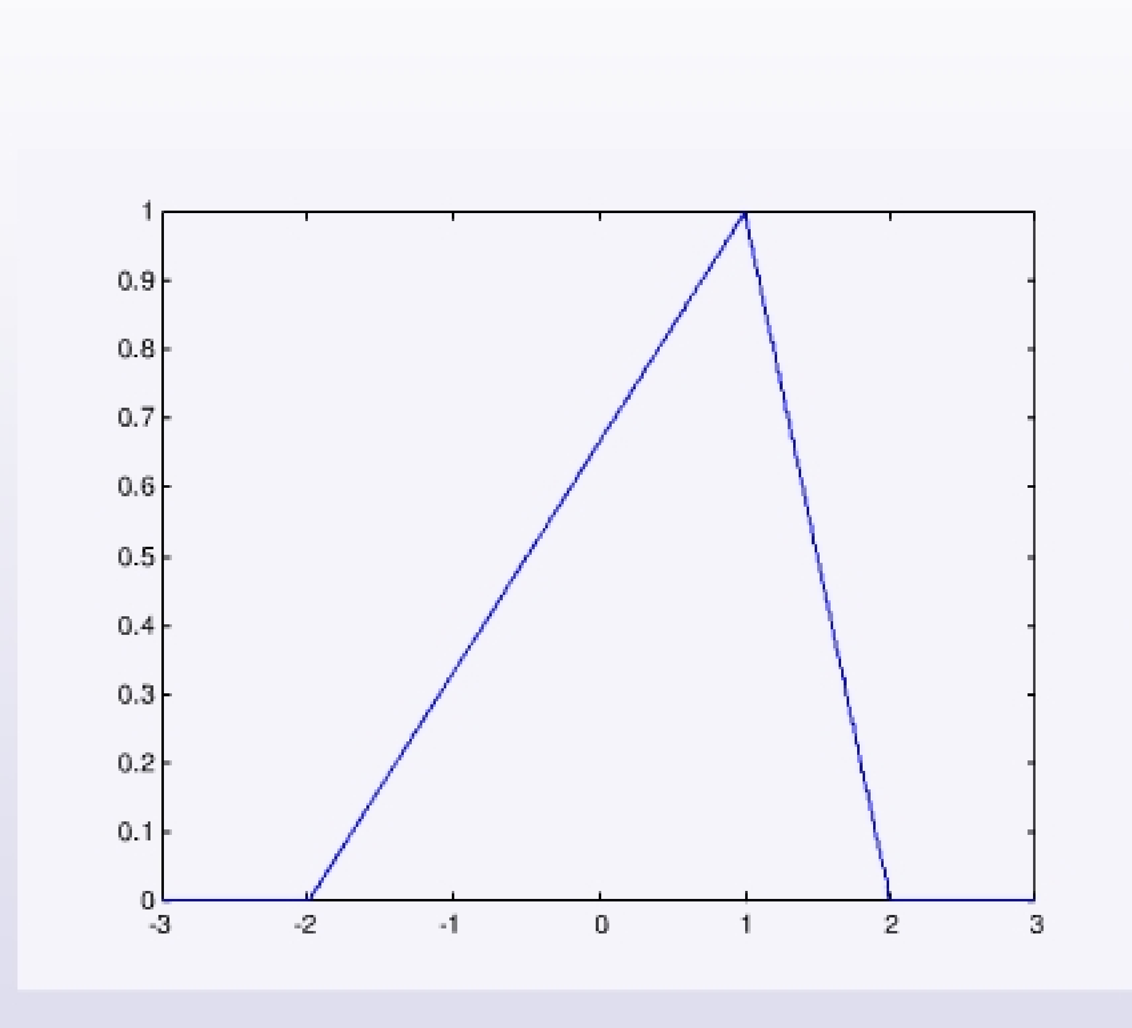
```
k0=0;k1=-50;k2=50;
[f,k]=stepseq(k0,k1,k2);
stem(k,f)
```

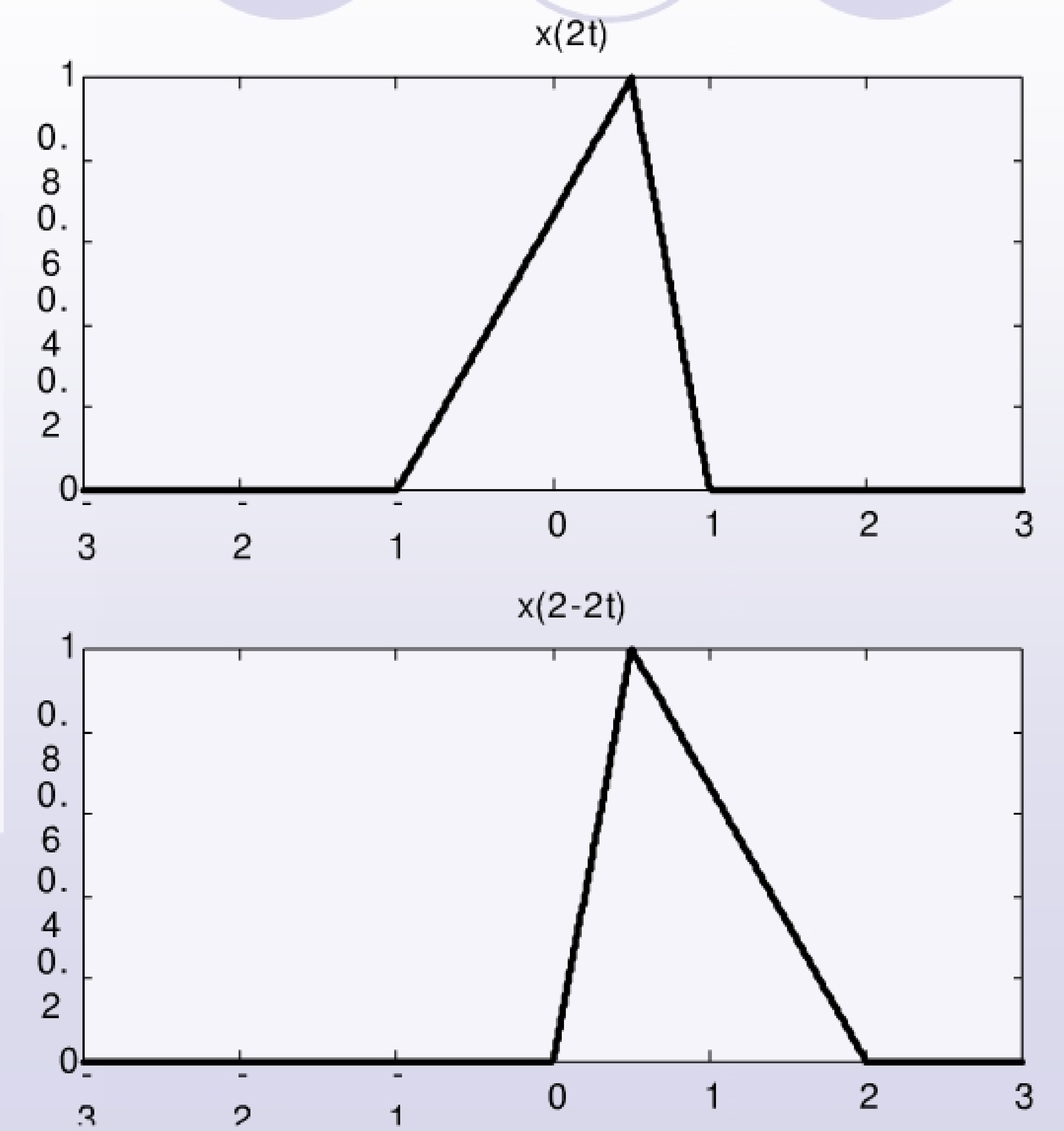
二、信号基本运算的MATLAB实现

1. 信号的尺度变换、翻转、时移（平移）

```
t=-3:0.001:3;
ft1=tripuls(2*t,4,0.5);
subplot(2,1,1)
plot(t,ft1)
title('x(2t)')
ft2=tripuls((2-2*t),4,0.5);
subplot(2,1,2)
plot(t,ft2)
title('x(2-2t)')
```

已知三角波$x(t)$，用MATLAB画出的$x(2t)$和$x(2-2t)$ 波形

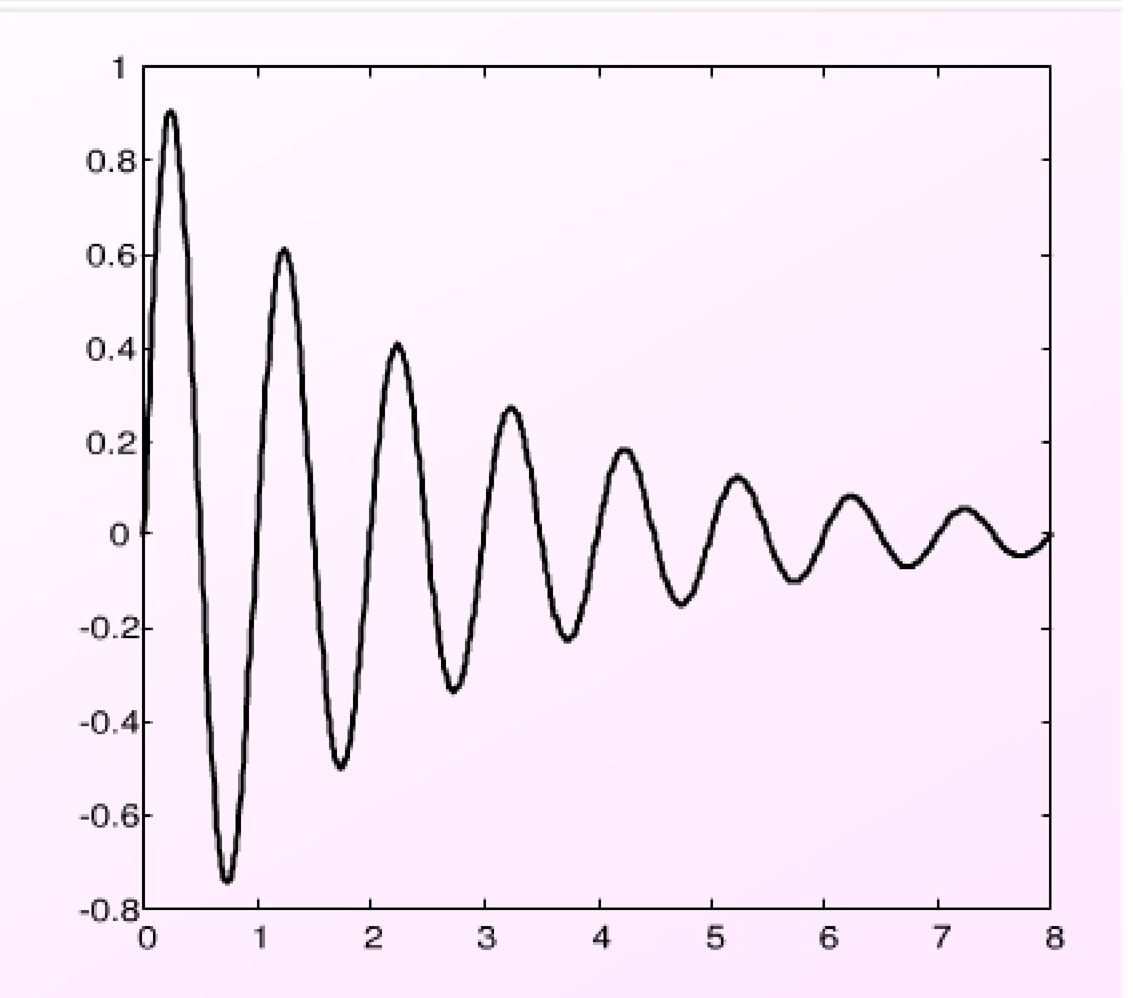




二、信号基本运算的MATLAB实现

2. 信号的相加与相乘

- ✓ 相加用算术运算符“+”实现
- ✓ 相乘用数组运算符“.*”实现



例:画信号$Ae^{at}\cos(\omega_0 t+\phi)$的波形

```
t=0:0.001:8;
A=1; a=-0.4;
w0=2*pi;phi=0;
ft1=A*exp(a*t).*sin(w0*t+phi);
plot(t,ft1)
```

二、信号基本运算的MATLAB实现

3. 离散序列的差分与求和
   连续信号的微分与积分

- 差分 y=diff(f);
- 求和 y=sum(f(k1:k2));

- 微分 y=diff(f)/h; h为数值计算所取时间间隔
- 定积分 quad('function_name',a,b);

function_name为被积函数名，a和b指定积分区间。

二、信号基本运算的MATLAB实现

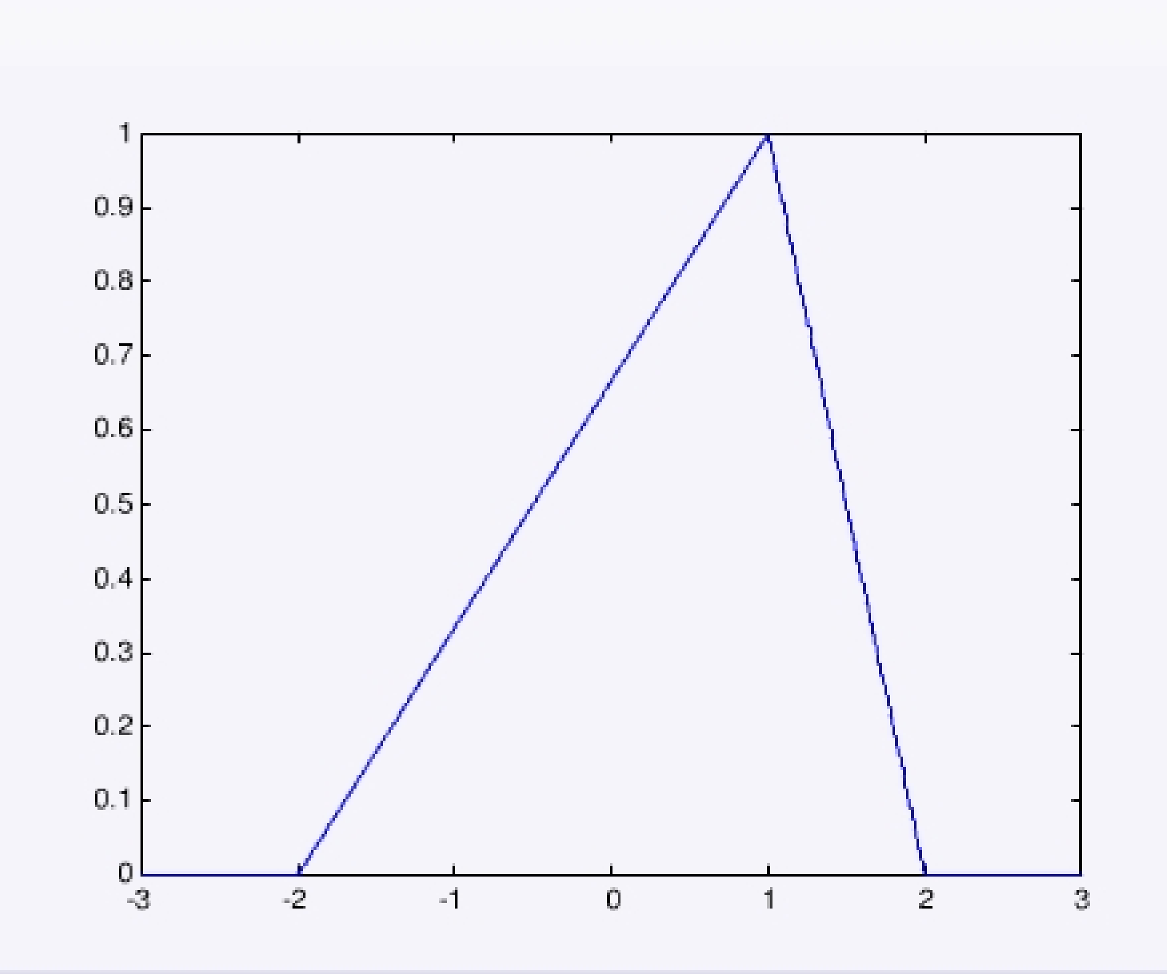
3. 离散序列的差分与求和
连续信号的微分与积分

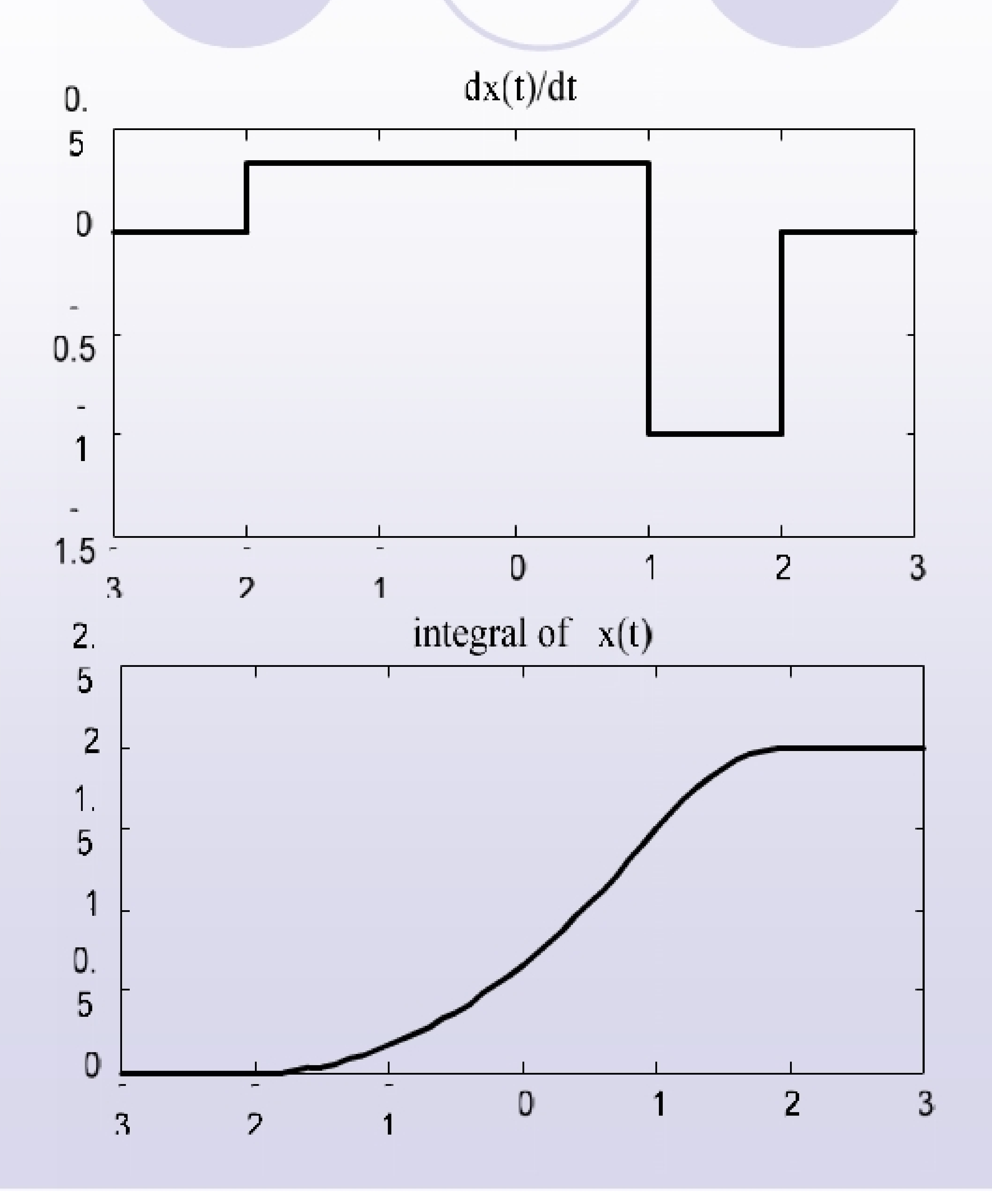
例:已知三角波$x(t)$，画出其微分与积分的波形

```
%differentiation
h=0.001;t= −3:h:3;
y1=diff(f2_2(t))*1/h;
plot(t(1:length(t)−1),y1)
```

```
%integration
t= −3:0.1:3;
for x=1:length(t)
  y2(x)=quad('f2_2', −3,t(x));
end
plot(t,y2)
```

三角波$f(t)$微分与积分的波形







利用MATLAB进行系统的时域分析

- 连续时间系统零状态响应的求解
- 连续时间系统冲激响应和阶跃响应的求解
- 离散时间系统零状态响应的求解
- 离散时间系统单位脉冲响应的求解
- 离散卷积的计算

一、连续时间系统零状态响应的求解

```
y=lsim(sys,x,t)
```

t 表示计算系统响应的抽样点向量

x 是系统输入信号向量，

sys 是LTI系统模型，借助tf函数获得

```
sys=tf(b,a)
```

b和a分别为微分方程右端和左端各项的系数向量

$$a_3y'''(t)+a_2y''(t)+a_1y'(t)+a_0y(t)=b_3x'''(t)+b_2x''(t)+b_1x'(t)+b_0x(t)$$

```
a=[a3, a2, a1, a0];
b=[b3, b2, b1, b0];
sys=tf(b,a)
```

二、连续系统冲激响应和阶跃响应求解

连续时间系统冲激响应可用impulse函数直接求出，其调用形式为

y=impulse(sys, t)

连续时间系统阶跃响应可用step函数直接求出，其调用形式为

y=step(sys, t)

t 表示计算系统响应的抽样点向量

sys 是LTI系统模型

三、离散时间系统零状态响应的求解

y=filter(b,a,x)

b , a 分别是差分方程左、右端的系数向量

x 表示输入序列， y 表示输出序列

$$\sum_{i=0}^{n} a_i y[k-i] = \sum_{j=0}^{m} b_j x[k-j]$$

可用MATLAB表示为

```
b=[b0,b1,b2,……,bM];
a=[a0,a1,a2, ……,aN];
```

四、离散时间系统单位脉冲响应的求解

h=impz(b,a,k)

b, a 分别是差分方程左、右端的系数向量

k 表示输出序列的取值范围

h 就是单位脉冲响应

五、离散卷积的计算

c=conv(a,b)

式中a,b为待卷积两序列的向量表示，c是卷积结果。

conv函数也可用于计算两个多项式的积

例：$(s^3+2s+3)(s^2+3s+2)$可用下面MATLAB语句求出

```
a =[1,0,2,3];
b =[1,3,2];
c=conv(a,b)
```

例1 求系统 $y''(t)+2y'(t)+100y(t)=10x(t)$ 的零状态响应，已知$x(t)=\sin(2\pi t)\,u(t)$。

```
%program3_1微分方程求解
ts=0;te=5;dt=0.01;
sys=tf([10],[1 2 100]);
t=ts:dt:te;
x=sin(2*pi*t);
y=lsim(sys,x,t);
plot(t,y);
xlabel('Time(sec)')
ylabel('y(t)')
```

例2 求系统 $y''(t)+2y'(t)+100y(t)=10x(t)$ 的零状态响应，已知$x(t)=\delta(t)$ 。

```
%program3_2连续时间系统的冲激响应
ts=0;te=5;dt=0.01;
sys=tf([10],[1 2 100]);
t=ts:dt:te;
y=impulse(sys,t);
plot(t,y);
xlabel('Time(sec)')
ylabel('h(t)')
```

例3 分析噪声干扰的信号$x[k]=s[k]+d[k]$通过M点滑动平均系统的响应，$y[k]=\frac{1}{M}\sum_{n=0}^{M-1}x[k-n]$

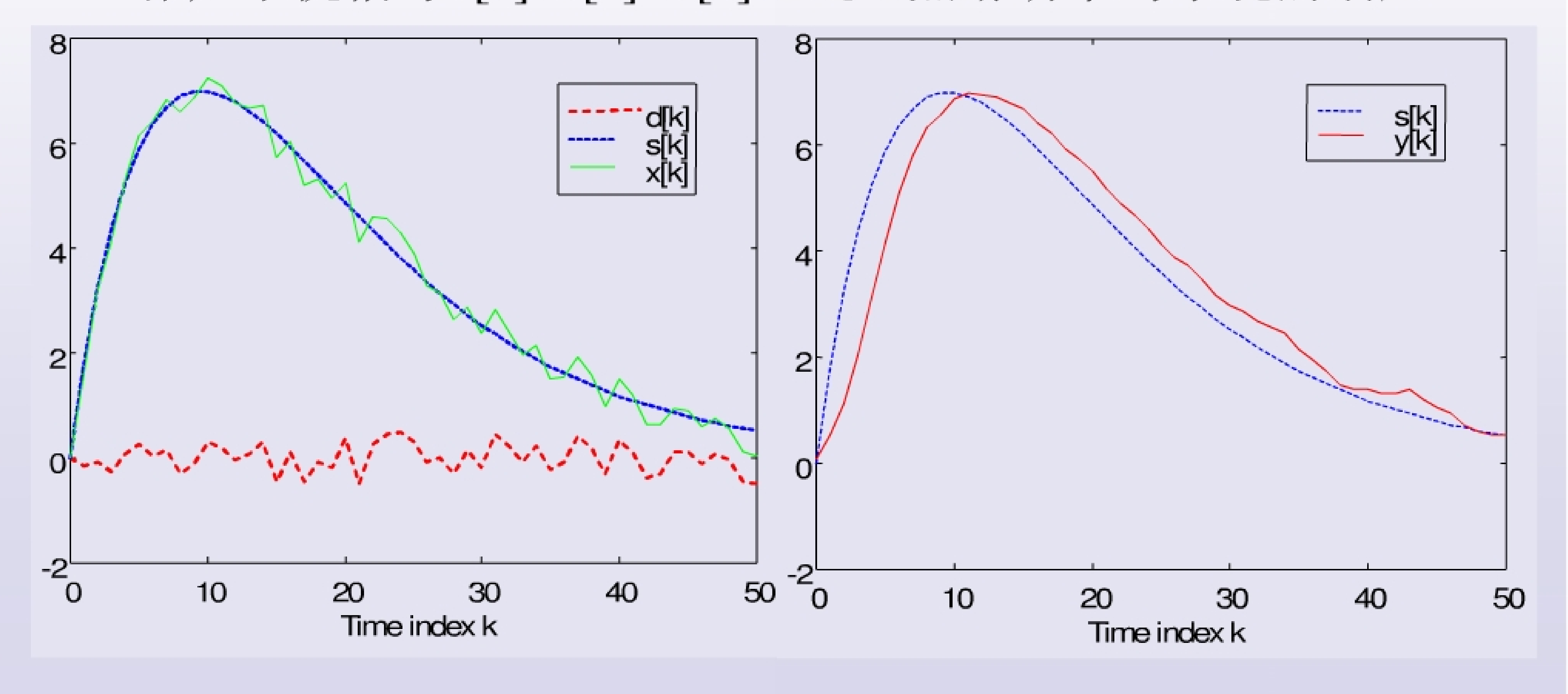
其中$s[k]=(2k)0.9^k$是原始信号，$d[k]$是噪声。

```
R =51 ; d = rand(1,R) − 0.5;
k=0:R−1;
s=2*k.*(0.9.^k); x=s+d;
figure(1);
plot(k,d,'r-.',k,s,'b--',k,x,'g-');
M =5; b = ones(M,1)/M; a = 1;
y = filter(b,a,x);
figure(2);
plot(k,s,'b--',k,y,'r-');
```

例3 分析噪声干扰的信号$x[k]=s[k]+d[k]$通过M点滑动平均系统的响应，$y[k]=\frac{1}{M}\sum_{n=0}^{M-1}x[k-n]$

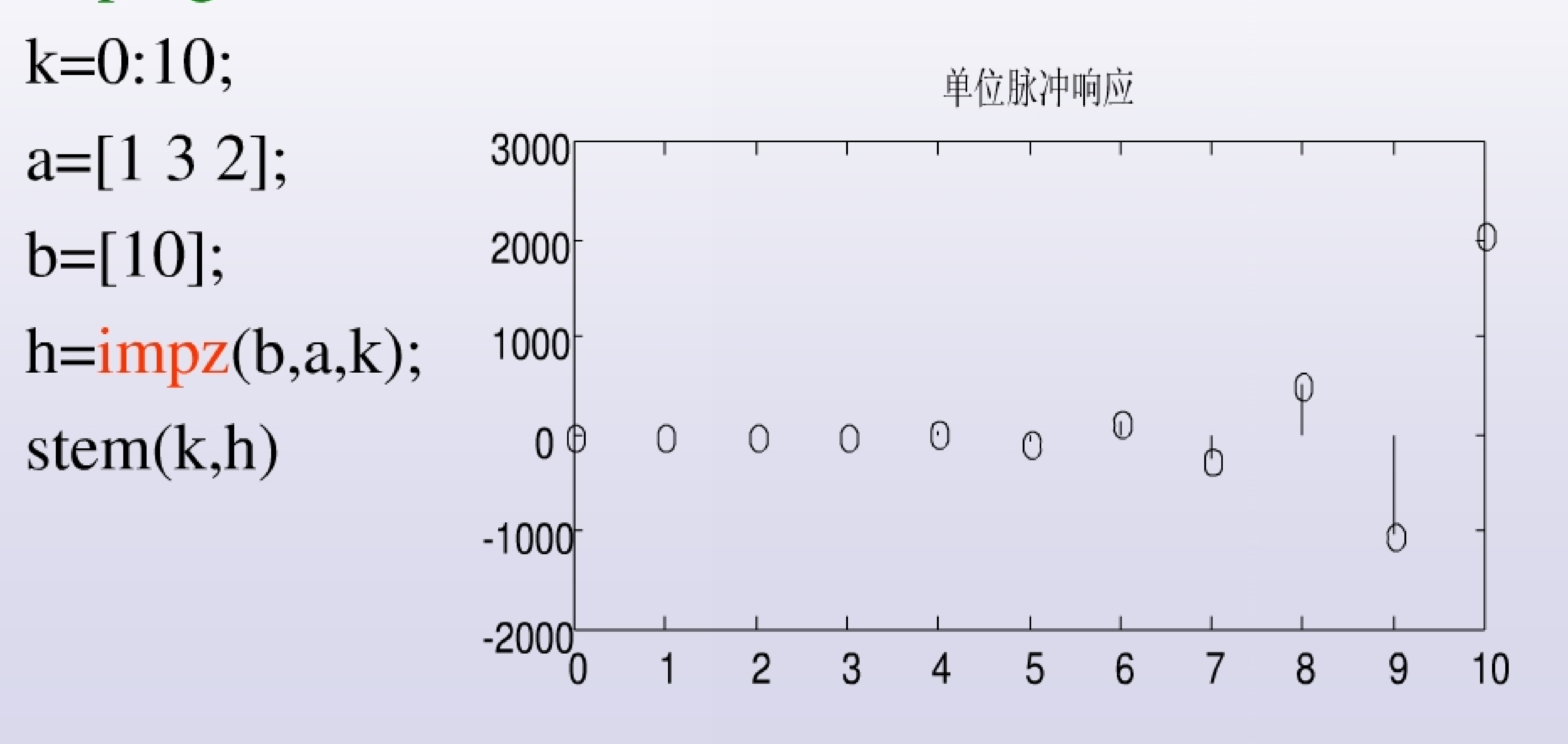
其中$s[k]=(2k)0.9^k$是原始信号，$d[k]$是噪声。

噪声干扰信号$x[k]=s[k]+d[k]$通过M点滑动平均系统的响应



例4 求系统$y[k]+3y[k-1]+2y[k-1]=10x[k]$的单位脉冲响应。

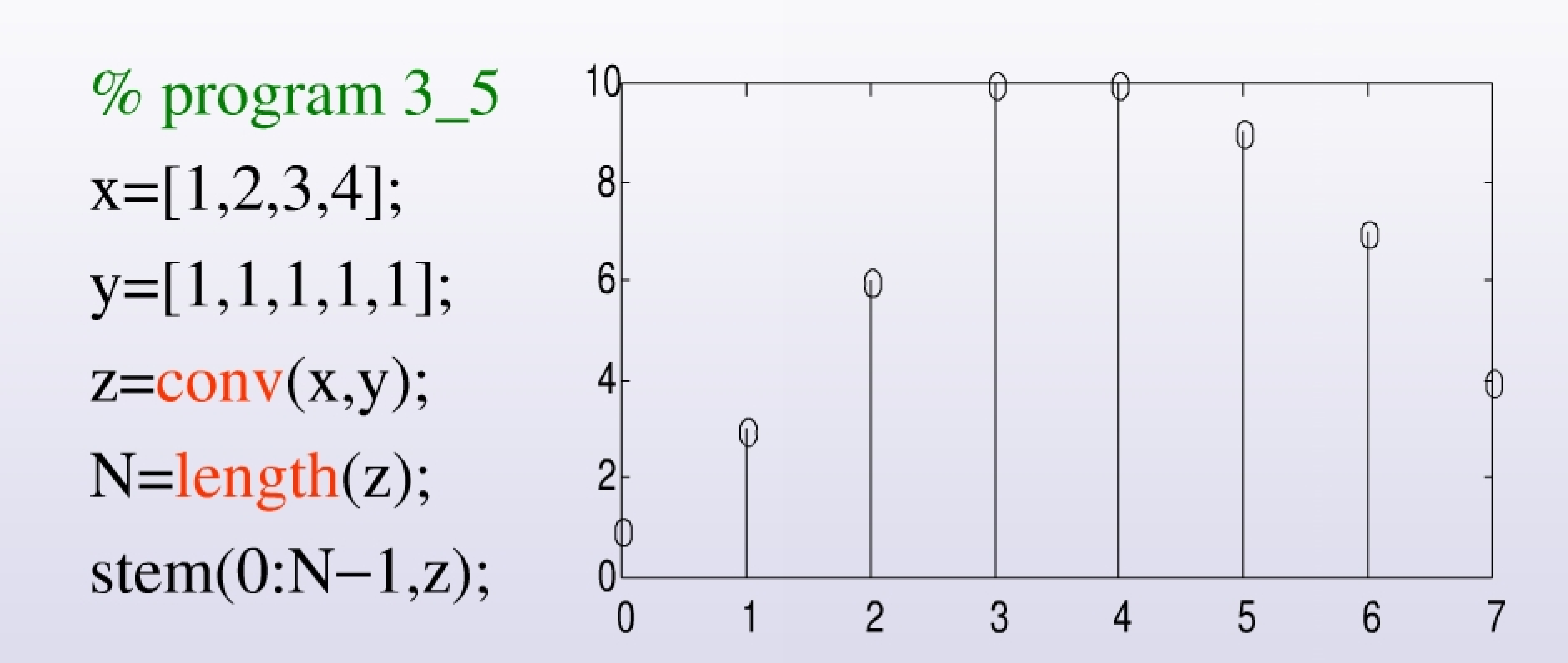
```
% program 3_4 离散系统的单位脉冲响应
k=0:10;
a=[1 3 2];
b=[10];
h=impz(b,a,k);
stem(k,h)
```



例**5** 计算$x[k]*y[k]$并画出卷积结果，已知

$x[k]=\{1,2,3,4;\ k=0,1,2,3\}$，

$y[k]=\{1,1,1,1,1;\ k=0,1,2,3,4\}$ 。





利用MATLAB进行信号的频域分析

- 周期信号频谱的**MATLAB**实现
- 用数值积分分析非周期信号频谱

一、周期信号频谱的MATLAB实现

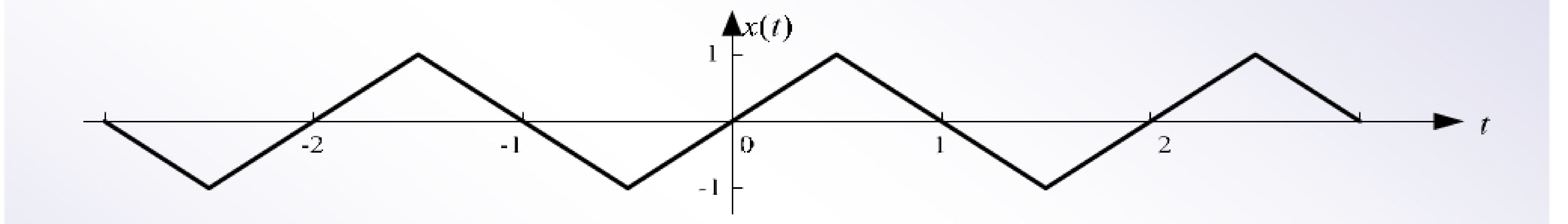
频谱Fn一般为复数，可分别利用abs和angle函数获得其幅频特性和相频特性。

其调用格式分别为

```
x=abs(Cn)
y=angle(Cn)
```

周期信号的频谱Cn 为离散信号，可以用stem画出其频谱图。

例**1** 试用MATLAB画出图示周期三角波信号的频谱。



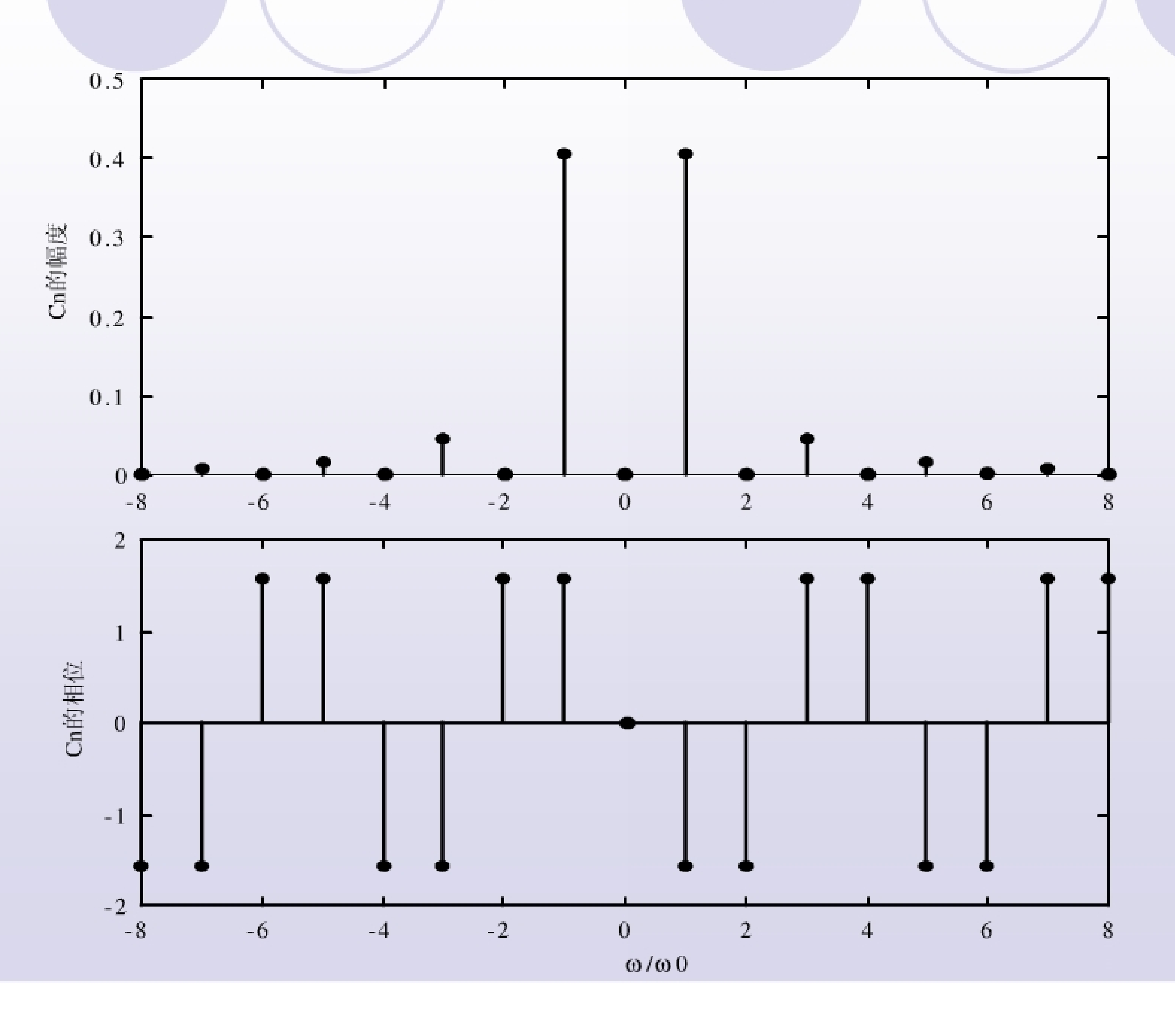
解：周期信号的频谱为

$$C_n = \begin{cases} \dfrac{-4\mathrm{j}}{n^2\pi^2}\sin(\dfrac{n\pi}{2}) & n \neq 0 \\ 0 & n = 0 \end{cases}$$

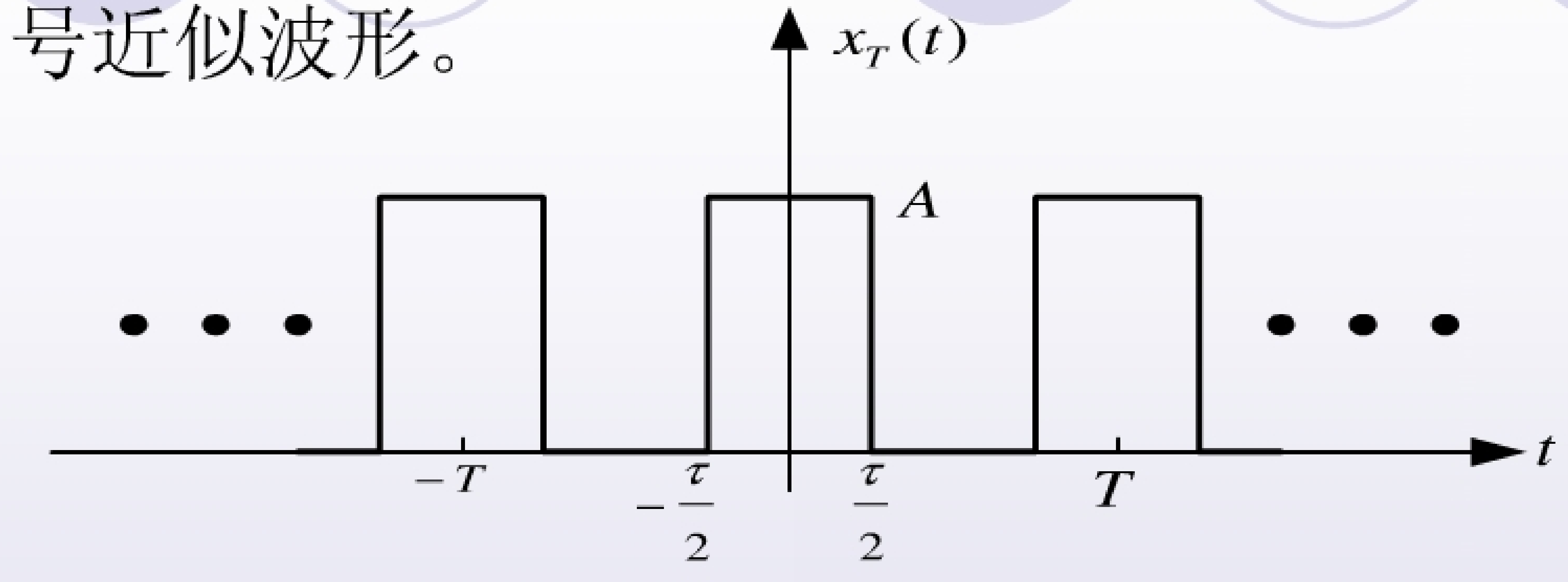
画三角波信号频谱的MATLAB程序

```
N=8;
n1= −N:−1; %计算n=−N到−1的Fourier系数
c1= −4*j*sin(n1*pi/2)/pi^2./n1.^2;
c0=0; %计算n=0时的Fourier系数
n2=1:N; %计算n=1到N的Fourier系数
c2= −4*j*sin(n2*pi/2)/pi^2./n2.^2;
cn=[c1 c0 c2];
n= −N:N;
subplot(2,1,1);
stem(n,abs(cn));ylabel('Cn的幅度');
subplot(2,1,2);
stem(n,angle(cn));
ylabel('Cn的相位');xlabel('\omega/\omega0');
```

程序运行结果



例**2** 求周期矩形脉冲的Fourier级数表示式。并用MATLAB求出由前N项Fourier级数系数得出的信号近似波形。



$$C_n = \frac{\tau A}{T}\mathrm{Sa}(n\omega_0\tau/2)$$

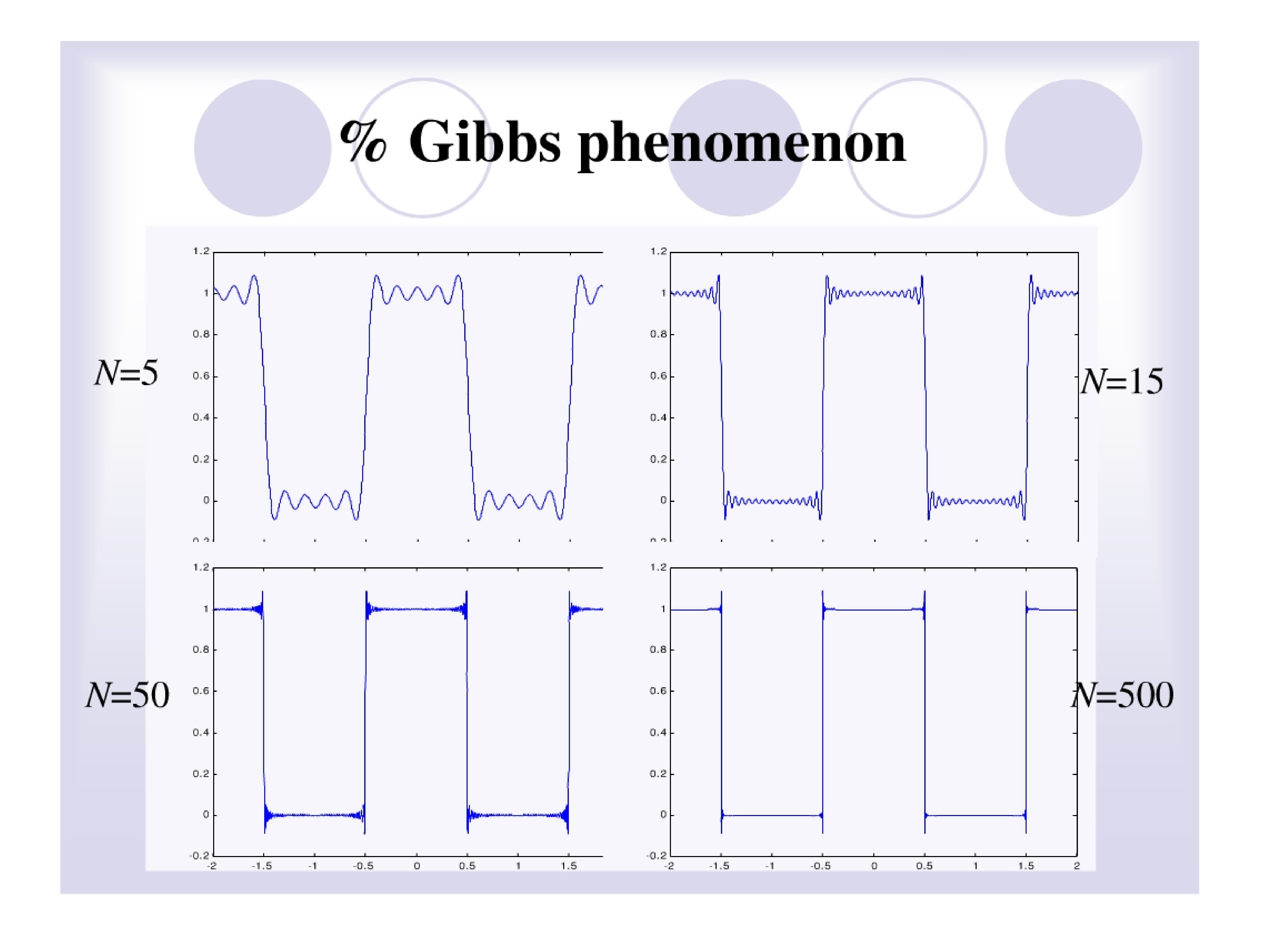
$$x_T(t) = \sum_{n=-\infty}^{\infty} C_n \mathrm{e}^{\mathrm{j}n\omega_0 t} = \frac{\tau A}{T} + \sum_{n=1}^{\infty} \frac{2\tau A}{T}\mathrm{Sa}(\frac{n\omega_0\tau}{2})\cos n\omega_0 t$$

取A=1, T=2, τ=1, ω_0=π

$$x_T(t) = 0.5 + \sum_{n=1}^{\infty} \mathrm{Sa}(\frac{n\pi}{2})\cos(n\pi t)$$

% Gibbs phenomenon

```
t=-2:0.001:2;
N=input('Number of harmonics= ');
c0=0.5;
xN=c0*ones(1,length(t));  %dc component
for n=0:2:N       % even harmonics are zero
      xN=xN+cos(pi*n*t)*sinc(n/2);
end
plot(t,xN);
```


% Gibbs phenomenon
N=5
N=15
N=50
N=500
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

二、用数值积分分析非周期信号频谱

数值函数积分quad8可用来计算非周期信号频谱

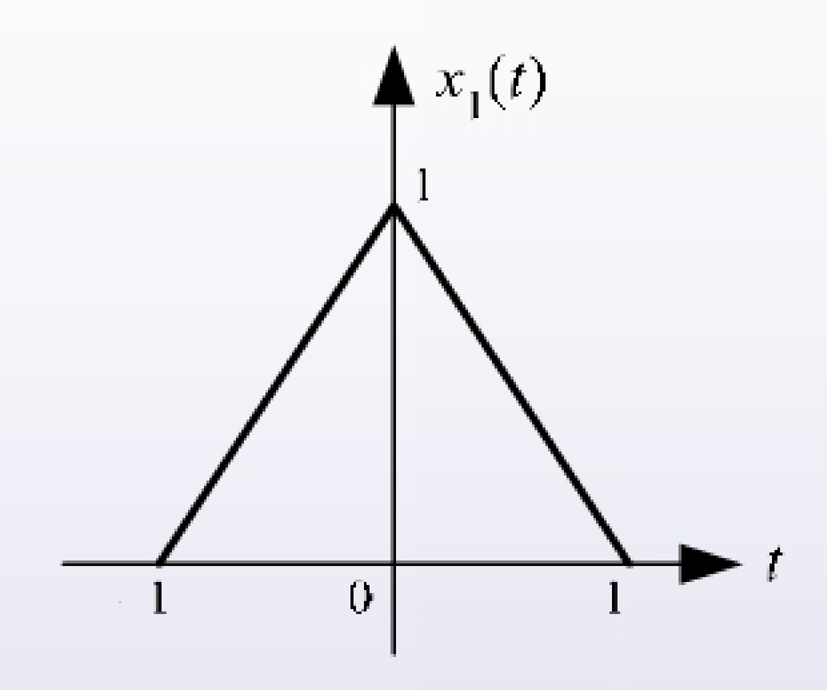
```
y = quad8('F',a,b)
```

F 是一个字符串，它表示被积函数的文件名。

a,b 分别表示定积分的下限和上限

quad8的返回是用自适应Simpson算法得出的积分值

例3　试用数值方法近似计算三角波信号的频谱



解： 图示三角波可表示为

$$x_1(t) = (1 - |t|)\, p_2(t)$$

三角波信号频谱的理论值为

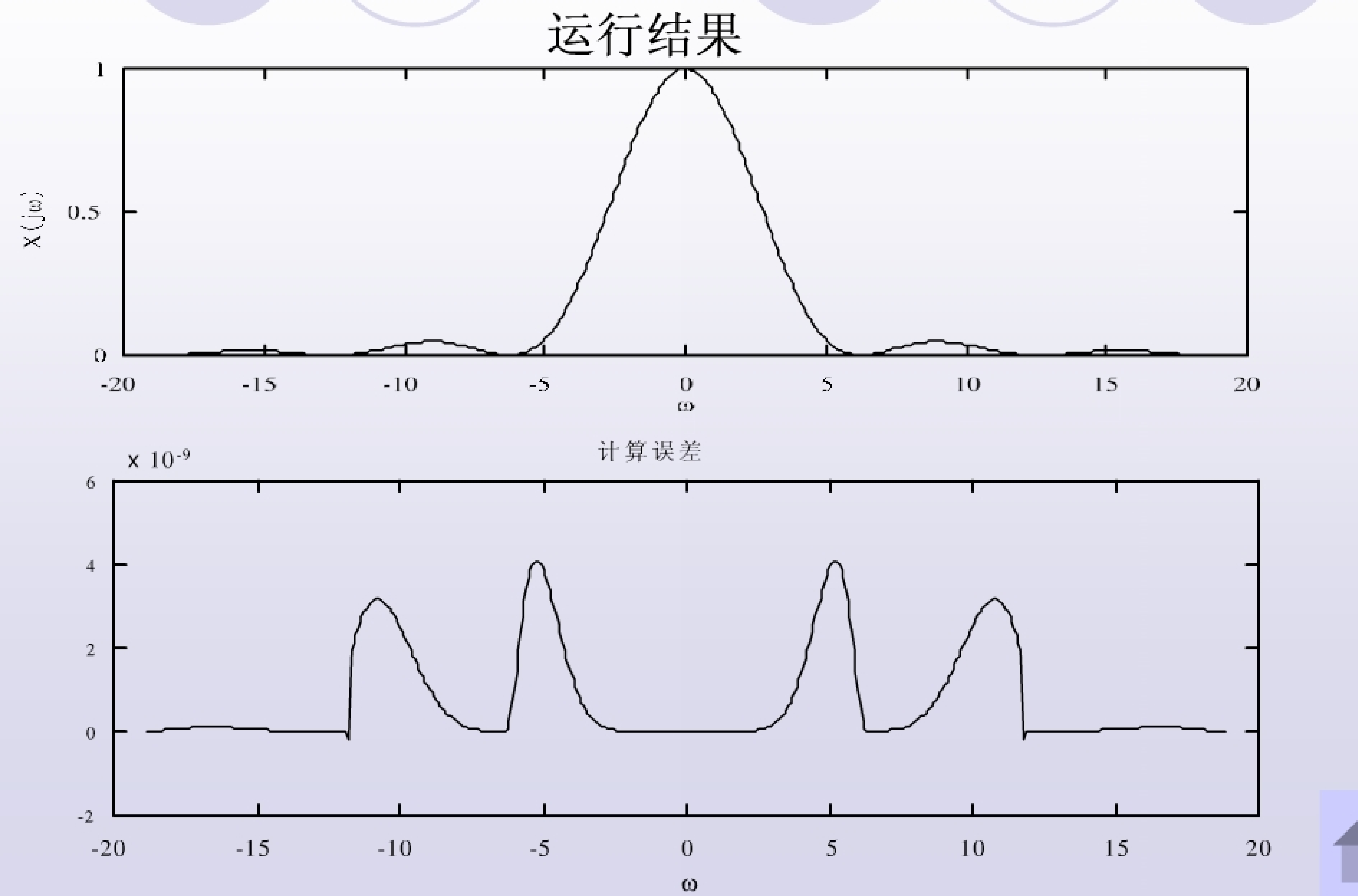
$$X(\mathrm{j}\omega) = \mathrm{Sa}^2(\omega / 2)$$

例3　试用数值方法近似计算三角波信号的频谱

```
function y=sf1(t,w);
y=(t>=-1 & t<=1).*(1-abs(t)).*exp(-j*w*t);
```

```
w=linspace(-6*pi,6*pi,512);
N=length(w);X=zeros(1,N);
for k=1:N
  X(k)=quad8('sf1',-1,1,[],[],w(k));
end
figure(1);
plot(w,real(X));title('')
xlabel('\omega');ylabel('X(j\omega)');
figure(2);
plot(w,real(X)-sinc(w/2/pi).^2);
xlabel('\omega');title('计算误差');
```

例3　试用数值方法近似计算三角波信号的频谱





利用MATLAB进行系统频域分析

- 连续系统频率响应的计算
- 周期信号通过系统的响应
- 离散系统频率响应的计算

一、连续系统频率响应的计算

$$H(\mathrm{j}\omega)=\frac{B(\omega)}{A(\omega)}=\frac{b(1)(\mathrm{j}\omega)^{M}+b(2)(\mathrm{j}\omega)^{M-1}+\cdots b(M+1)}{a(1)(\mathrm{j}\omega)^{N}+a(2)(\mathrm{j}\omega)^{N-1}+\cdots a(N+1)}$$

计算频响的MATLAB函数

```
H=freqs(b,a,w)
```

b　分子多项式系数
a　分母多项式系数
w　需计算的$H(\mathrm{j}\omega)$的抽样点
　(数组w中少需包含两个ω的抽样点)。

一、连续系统频响特性的计算

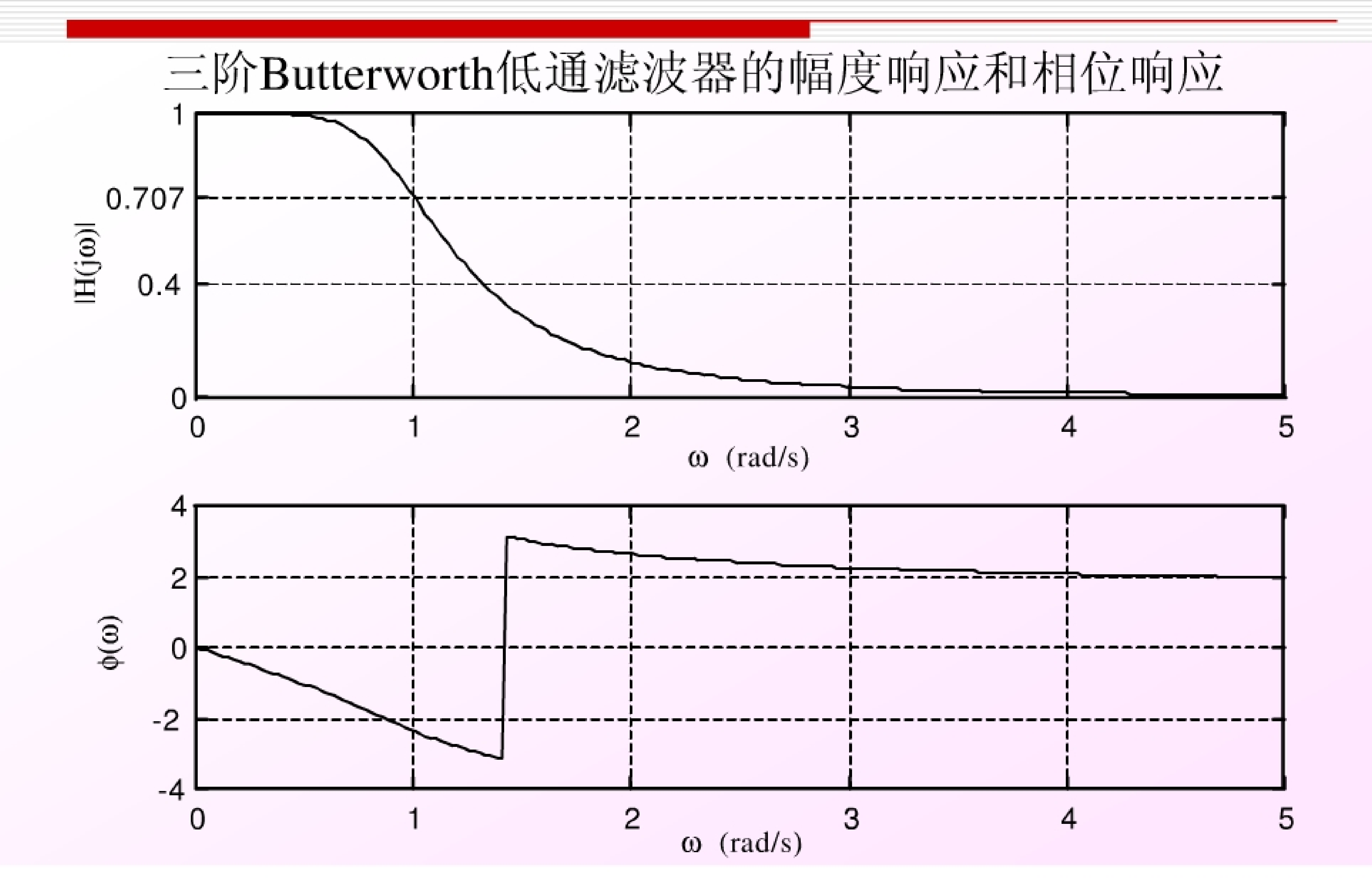
例**1** 三阶归一化的Butterworth低通滤波器的系统函数为

$$H(\mathrm{j}\omega)=\frac{1}{(\mathrm{j}\omega)^3+2(\mathrm{j}\omega)^2+2(\mathrm{j}\omega)+1}$$

试画出$|H(\mathrm{j}\omega)|$ 和$\varphi(\omega)$。

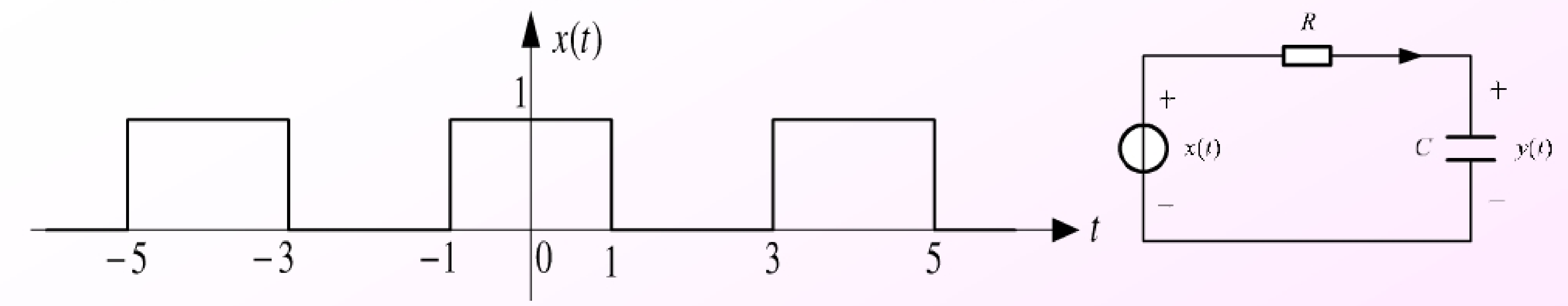
```
w=linspace(0,5,200);
b=[1];a=[1 2 2 1];
h=freqs(b,a,w);
subplot(2,1,1);
plot(w,abs(h));
subplot(2,1,2);
plot(w,angle(h));
```

一、连续系统频响特性的计算



二、周期信号通过系统的响应

例**2**　周期方波通过**RC**系统的响应。



$$H(\mathrm{j}\omega)=\frac{1/\mathrm{j}\omega C}{R+1/\mathrm{j}\omega C}=\frac{1}{1+\mathrm{j}\omega RC}\qquad C_n=\frac{A\tau}{T}\mathrm{Sa}\left(\frac{n\omega_0\tau}{2}\right)=\frac{1}{2}\mathrm{Sa}\left(\frac{\pi n}{2}\right)$$

$$y(t)=\sum_{n=-\infty}^{\infty}C_nH(\mathrm{j}n\omega_0)\mathrm{e}^{\mathrm{j}n\omega_0 t}=0.5+\sum_{n=1}^{\infty}\mathrm{Sa}(\frac{\pi n}{2})\left|\frac{1}{1+\mathrm{j}n\omega_0 RC}\right|\cos(n\omega_0 t+\phi)$$

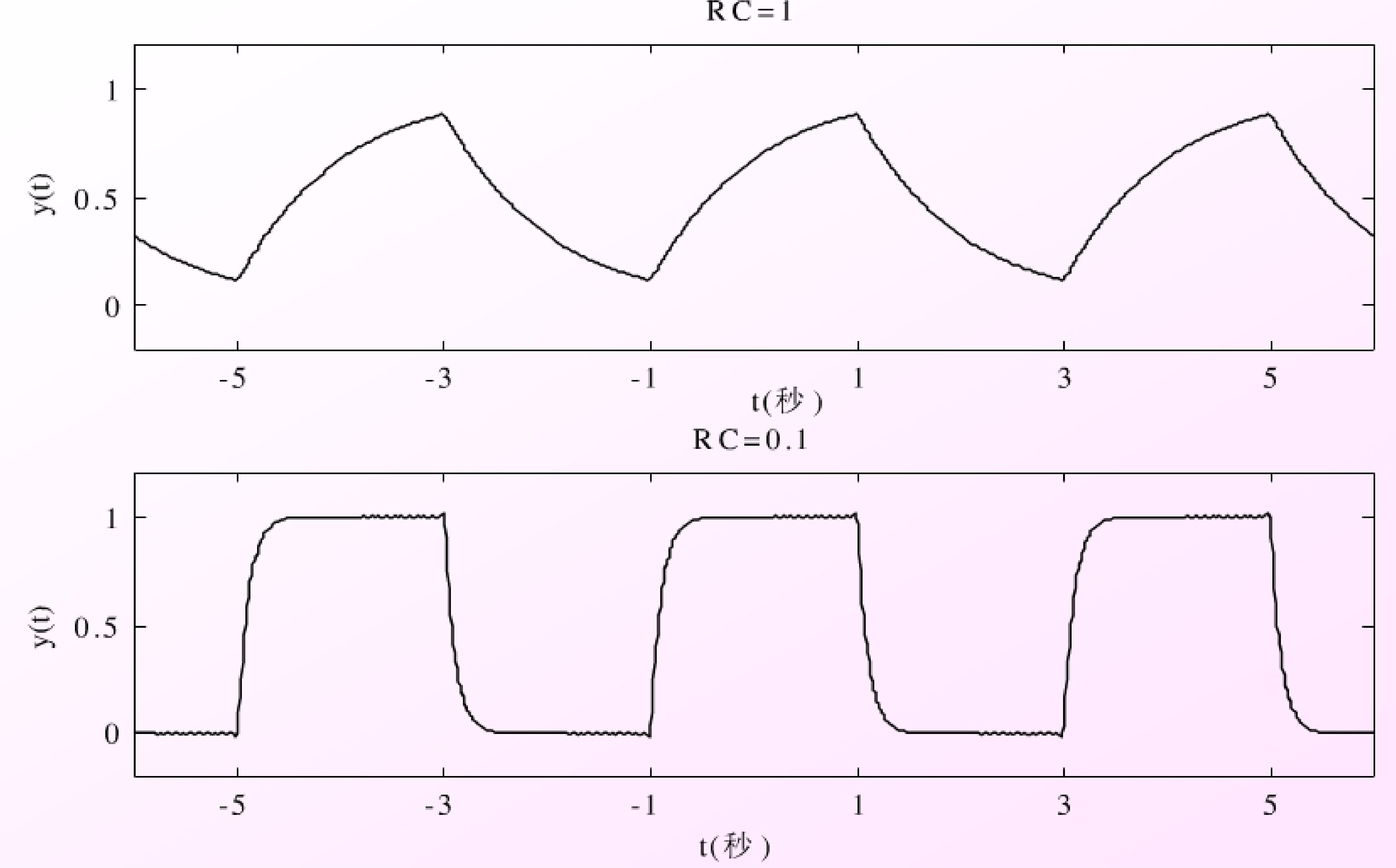
二、周期信号通过系统的响应

例**2**　周期方波通过**RC**系统的响应。

```
%p5_2 Periodic signal pass LTI system
T=4;w0=2*pi/T;RC=0.1;
t= –6:0.01:6;N=51;
c0=0.5;xN=c0*ones(1,length(t)); %dc
for n=1:2:N   % even harmonics are zero
  H=abs(1/(1+j*RC*w0*n));
  phi=angle(1/(1+j*RC*w0*n));
  xN=xN+H*cos(w0*n*t+phi)*sinc(n*0.5);
end
plot(t,xN);
xlabel(['time RC=',num2str(RC)]);grid;
set(gca,'xtick',[–5  –3  –1  0  1  3  5]);
```

二、周期信号通过系统的响应

例**2**　周期方波通过**RC**系统的响应。



三、离散系统频率响应的计算

$$H(\mathrm{e}^{\mathrm{j}\Omega}) = \frac{B(\mathrm{e}^{\mathrm{j}\Omega})}{A(\mathrm{e}^{\mathrm{j}\Omega})} = \frac{b_0 + b_1\mathrm{e}^{-\mathrm{j}\Omega} + \cdots + b_M\mathrm{e}^{-\mathrm{j}\Omega M}}{a_0 + a_1\mathrm{e}^{-\mathrm{j}\Omega} + \cdots + a_N\mathrm{e}^{-\mathrm{j}\Omega N}}$$

计算频率响应的MATLAB函数

```
h = freqz(b,a,w)
```

b　分子的系数　　　a　分母系数

w　抽样的频率点(至少2点)， w在0~2π之间

幅频特性：abs，相频特性: angle

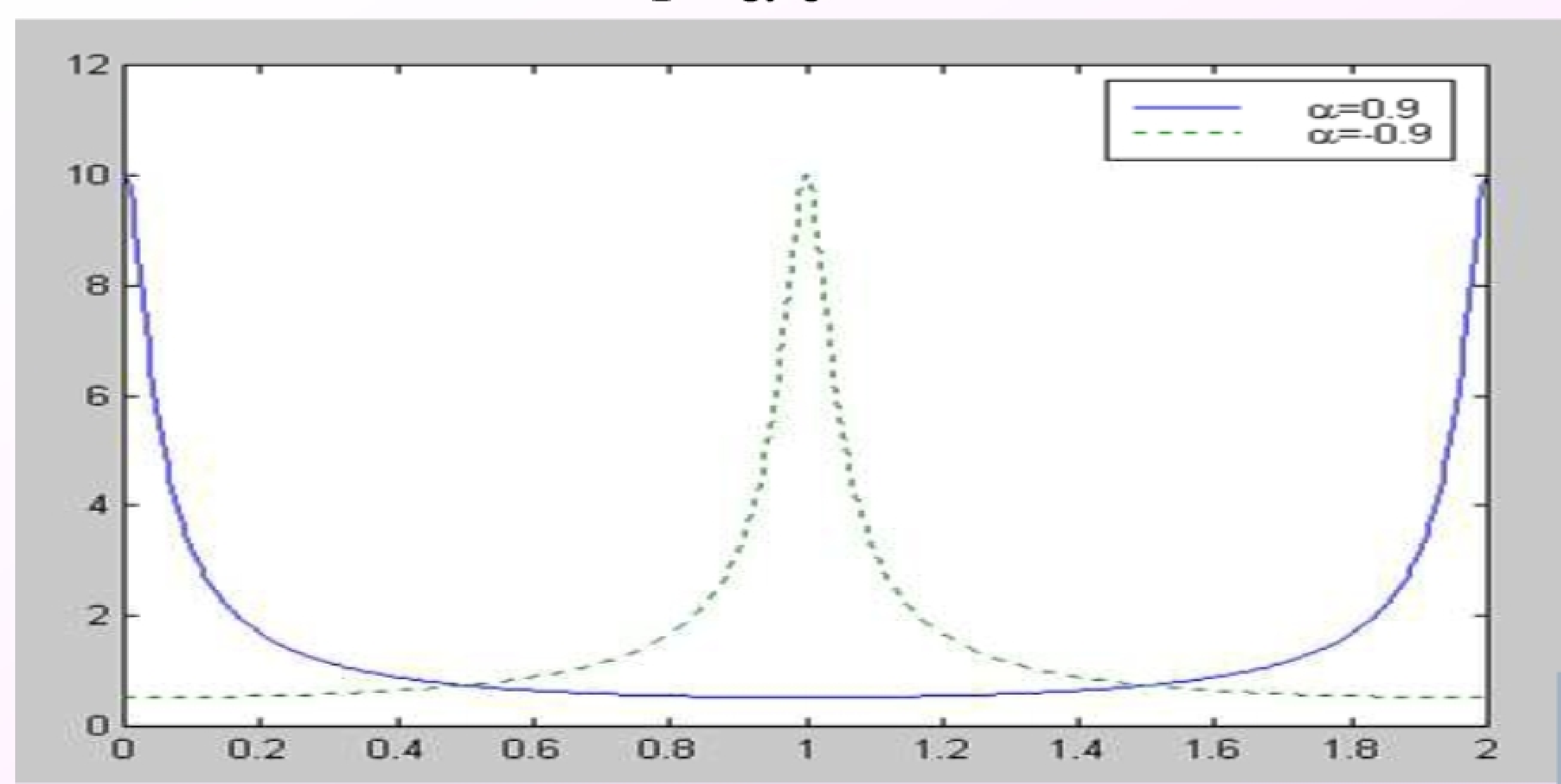
三、离散系统频率响应的计算

例：画 $H(\mathrm{e}^{\mathrm{j}\Omega}) = \dfrac{1}{1-\alpha\,\mathrm{e}^{-\mathrm{j}\Omega}}$ 的幅度响应曲线

```
b=[1];
a1=[1 -0.9]; a2=[1 0.9];
w=linspace(0,2*pi,512);
h1=freqz(b,a1,w);
h2=freqz(b,a2,w);
plot(w/pi,abs(h1),w/pi,abs(h2),':');
legend('\alpha=0.9','\alpha=-0.9');
```

三、离散系统频率响应的计算

例：画$H(\mathrm{e}^{\mathrm{j}\Omega})=\dfrac{1}{1-\alpha\,\mathrm{e}^{-\mathrm{j}\Omega}}$的幅度响应曲线





利用MATLAB进行连续系统的复频域分析

- 部分分式展开的**MATLAB**实现
- $H(s)$的零极点与系统特性的**MATLAB**计算

一、部分分式展开的MATLAB实现

[r,p,k]=residue(num,den)

num,den分别为$X(s)$分子多项式和分母多项式的系数向量。

r为部分分式的系数，p为极点，k为多项式的系数。若为真分式，则k为零。

二、$H(s)$的零极点与系统特性的MATLAB计算

计算多项式根roots的函数可用于计算$H(s)$的零极点。

```
r=roots(D) %计算多项式D(s)的根
```

$H(s)$零极点分布图可用pzmap函数画出，调用形式为

```
pzmap(sys)
```

表示画出sys所描述系统的零极点图。

例**1**　用部分分式展开法求$X(s)$的反变换。

$$X(s)=\frac{s+2}{s^3+4s^2+3s}$$

```
%program7_1
format rat %将结果数据以分数的形式输出
num=[1 2];  den=[1 4 3 0];
[r,p]=residue(num,den)
```

运行结果为　r = −1/6 ,−1/2 ,2/3

p = −3 ,−1 ,0

故$F(s)$可展开为　$X(s)=\dfrac{2/3}{s}+\dfrac{-0.5}{s+1}+\dfrac{-1/6}{s+3}$

$$x(t)=L^{-1}[X(s)]=\frac{2}{3}u(t)-\frac{1}{2}e^{-t}u(t)-\frac{1}{6}e^{-3t}u(t)$$

例**2** 用部分分式展开法求$X(s)$的反变换。

$$X(s)=\frac{2s^3+3s^2+5}{(s+1)(s^2+s+2)}$$

```
% program7_2
num=[2 3 0 5];
den=conv([1 1],[1 1 2]);
%将因子相乘的形式转换成多项式的形式
[r,p,k]=residue(num,den)
magr=abs(r) %求r的模
angr=angle(r) %求r的相角
```

例2　用部分分式展开法求$X(s)$的反变换。

$$X(s)=\frac{2s^3+3s^2+5}{(s+1)(s^2+s+2)}$$

运行结果为

- r =−2.0000 + 1.1339i, −2.0000 − 1.1339i, 3.0000
- p =−0.5000 + 1.3229i, −0.5000 − 1.3229i, −1.0000
- k =2
- magr =2.299, 2.2991, 3.0000
- angr =2.6258, −2.6258, 0

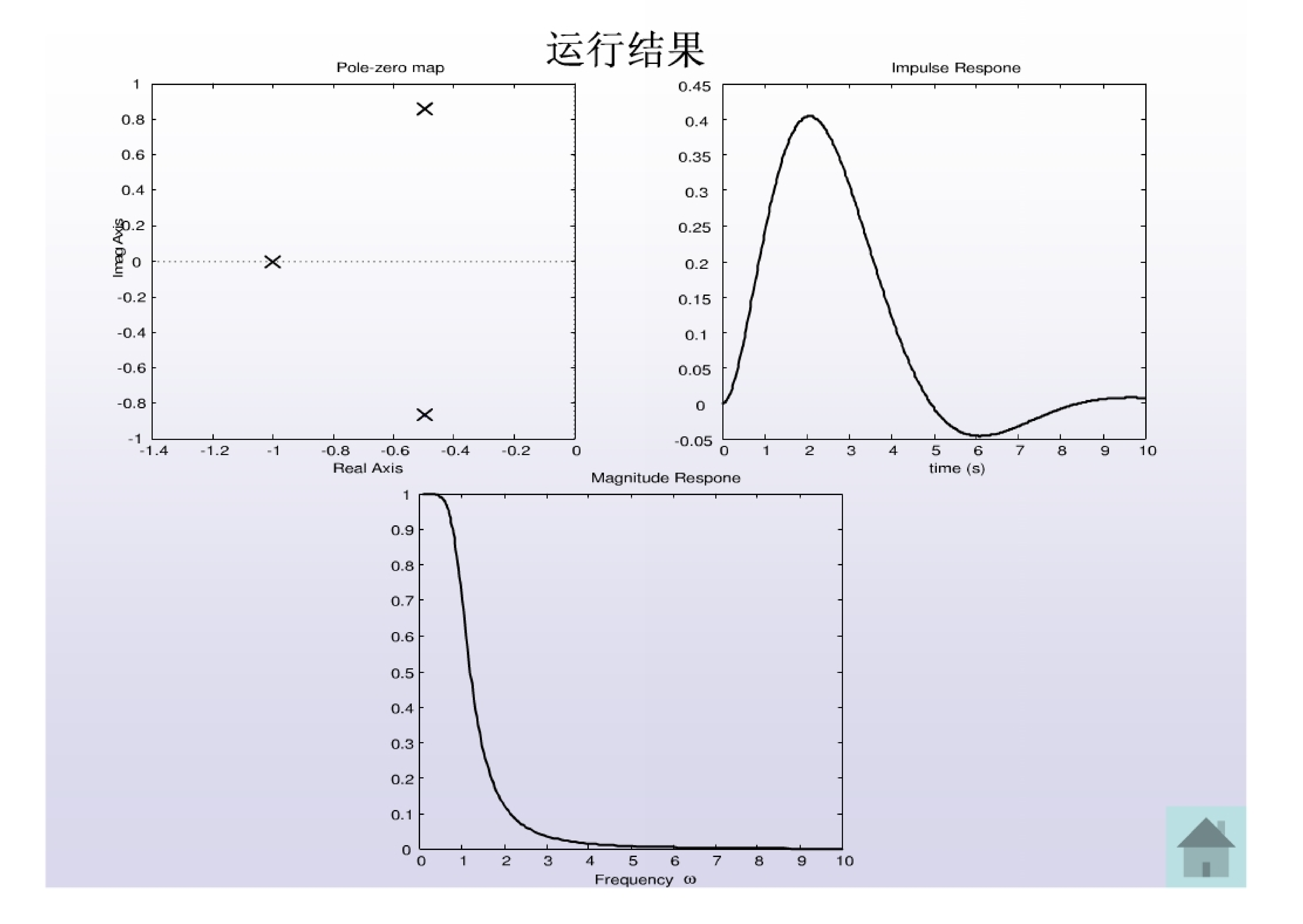
故$F(s)$可展开为

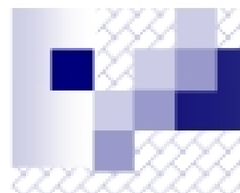
$$X(s)=2+\frac{3}{s+1}+\frac{2.2991\mathrm{e}^{-\mathrm{j}2.6258}}{s+0.5+\mathrm{j}1.3229}+\frac{2.2991\mathrm{e}^{\mathrm{j}2.6258}}{s+0.5-\mathrm{j}1.3229}$$

$$x(t)=2\delta(t)+3\mathrm{e}^{-t}u(t)+1.1495\mathrm{e}^{-0.5t}\cos(1.3229t+2.6258)u(t)$$

例3 试画出系统 $H(s)=\dfrac{1}{s^3+2s^2+2s+1}$ 的零极点分布图，求其单位冲激响应$h(t)$和频率响应$H(\mathrm{j}\omega)$，并判断系统是否稳定。

```
num=[1];den=[1 2 2 1];
sys=tf(num,den);
poles=roots(den)
figure(1);pzmap(sys);
t=0:0.02:10;
h=impulse(num,den,t);
figure(2);plot(t,h)
title('Impulse Respone')
[H,w]=freqs(num,den);
figure(3);plot(w,abs(H))
xlabel('\omega')
title('Magnitude Respone')
```


运行结果
Pole-zero map
Imag Axis
Real Axis
Impulse Respone
time (s)
Magnitude Respone
Frequency ω



利用MATLAB进行离散系统的z域分析

- 部分分式展开的**MATLAB**实现
- $H(z)$的零极点与系统特性的**MATLAB**计算

一、部分分式展开的MATLAB实现

```
[r,p,k]=residuez(num,den)
```

num,den分别为$X(z)$分子多项式和分母多项式的系数向量。

r为部分分式的系数，p为极点，k为多项式的系数。若为真分式，则k为零。

二、$H(z)$的零极点与系统特性的MATLAB计算

利用tf2zp函数计算$H(z)$的零极点，调用形式为

```
[z,p,k]=tf2zp(b,a)
```

b和a分别为$H(z)$分子多项式和分母多项式的系数向量。

返回值z为零点、p为极点、 k为增益常数。

$H(z)$零极点分布图可用zplane函数画出，

调用形式为

```
zplane(b,a)
```

例1 将$X(z)$用部分分式展开。

$$X(z)=\frac{18}{18+3z^{-1}-4z^{-2}-z^{-3}}$$

```
%program8_1
num = [18]; den = [18 3 −4 −1];
[r,p,k] = residuez(num,den)
```

运行结果为

r =0.3600，0.2400，0.4000

p =0.5000，−0.3333，−0.3333

k =[]

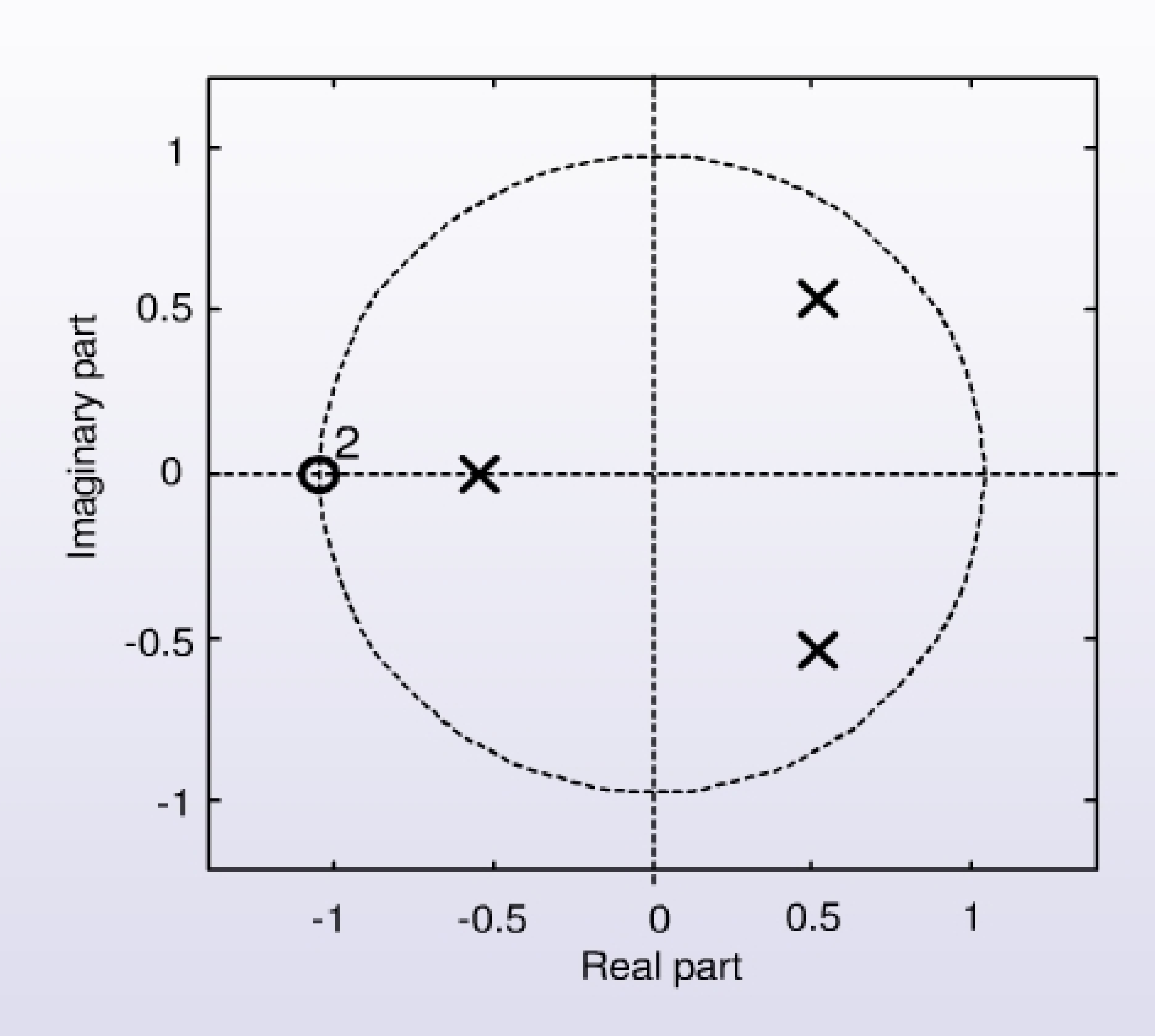
故$X(z)$可展开为

$$X(z)=\frac{0.36}{1-0.5z^{-1}}+\frac{0.24}{1+0.3333z^{-1}}+\frac{0.4}{(1+0.3333z^{-1})^{2}}$$

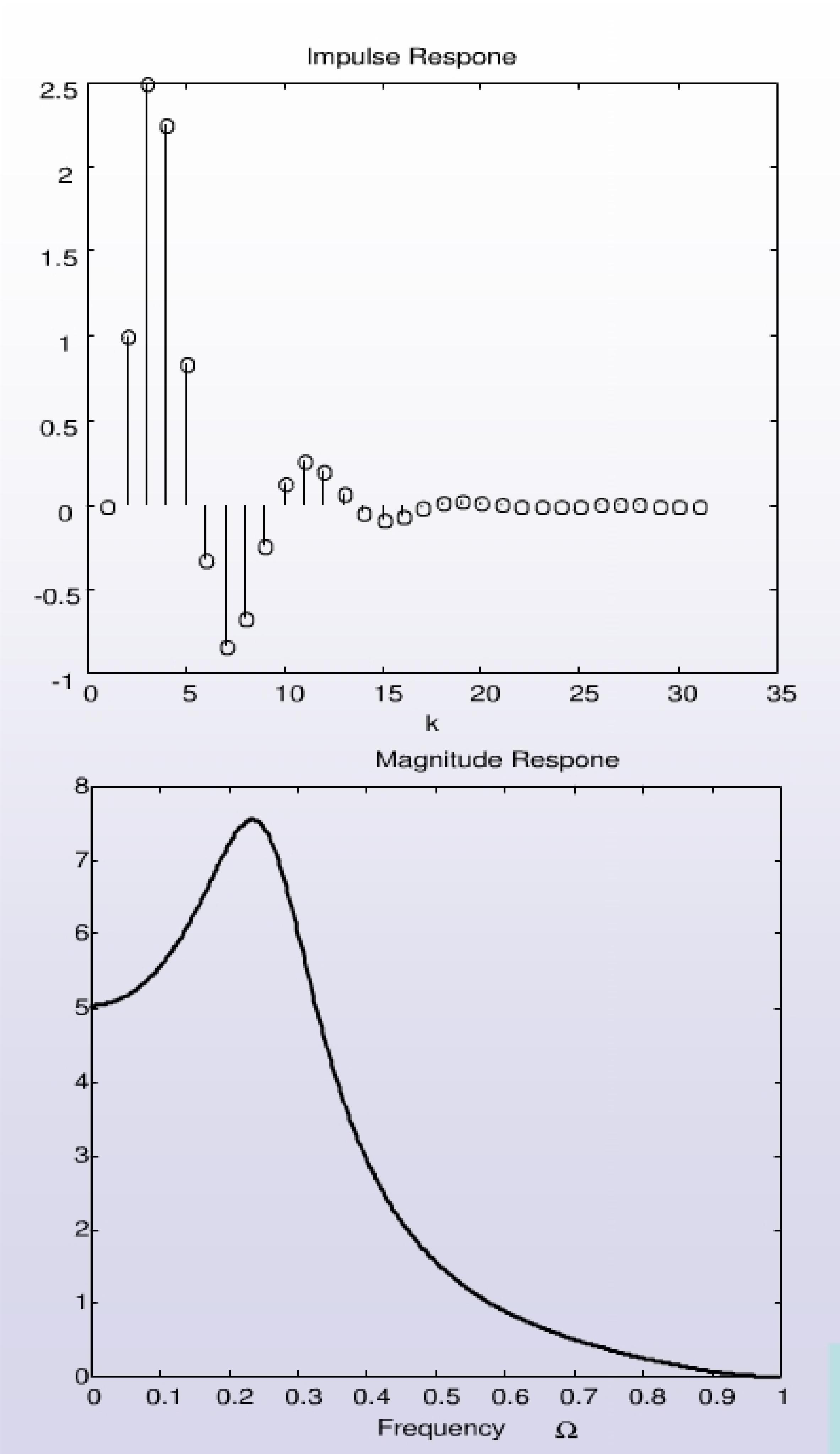
例2 试画出系统 $H(z)=\dfrac{z^{-1}+2z^{-2}+z^{-3}}{1-0.5z^{-1}-0.005z^{-2}+0.3z^{-3}}$ 的零极点分布图，求其单位冲激响应$h[k]$和频率响应$H(e^{j\Omega})$ 。

```
% program 8_2
b =[1 2 1];a =[1 –0.5 –0.005 0.3];
figure(1);zplane(b,a);
num=[0 1 2 1];
den=[1 –0.5 –0.005 0.3];
h=impz(num,den);
figure(2);stem(h)
xlabel('k')
title('Impulse Respone')
[H,w]=freqz(num,den);
figure(3);plot(w/pi,abs(H))
xlabel('Frequency \omega')
title('Magnitude Respone')
```

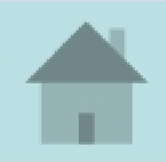
运行结果



Imaginary part
Real part
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1
2



Impulse Respone
k
Magnitude Respone
Frequency Ω



利用MATLAB进行系统状态变量分析

- 微分方程到状态方程的转换
- 状态方程系统函数矩阵$H(s)$的计算
- **MATLAB**求解连续系统的状态方程
- **MATLAB**求解离散系统的状态方程

一、微分方程到状态方程的转换

[A,B,C,D]= tf2ss(num,den)

num,den 分别表示系统函数$H(s)$的分子和分母多项式；

A,B,C,D 分别为状态方程的矩阵。

二、状态方程系统函数矩阵$\boldsymbol{H}(s)$的计算

[num,den]= ss2tf (A,B,C,D,k)

A, B, C, D 分别表示状态方程的矩阵。

K 表示函数ss2tf计算的与第k个输入相关的系统函数，即$\boldsymbol{H}(s)$的第k列。

num 表示$\boldsymbol{H}(s)$第k列的m个元素的分子多项式

den 表示$\boldsymbol{H}(s)$公共的分母多项式。

三、MATLAB求解连续系统的状态方程

获得连续系统状态方程的计算机表示模型

```
sys = ss(A,B,C,D)
```

求解状态方程

```
[y,to,q]=lsim(sys,x,t,q0)
```

sys 由函数ss构造的状态方程模型

t 需计算的输出样本点, t=0:dt:Tfinal

x(:,k) 系统第k个输入在t上的抽样值

q0 系统的初始状态(可缺省)

y(:,k) 系统的第k个输出

to 实际计算时所用的样本点； q 系统的状态

四、MATLAB求解离散系统的状态方程

获得离散状态方程的计算机表示模型

```
sys = ss(A,B,C,D,[ ])
```

求解状态方程

```
[y,n,q]=lsim(sys,x,[ ],q0)
```

sys 由函数ss构造的状态方程模型

x(:,k) 系统第k个输入序列

q0 系统的初始状态(可缺省)

y(:,k) 系统的第k个输出

n 序列的下标; q 系统的状态

或直接利用

```
[y, q]=dlsim(A,B,C,D,x,q0)
```

例1 写出系统 $y''(t)+5y'(t)+10y(t)=x(t)$ 的状态方程。

由 [A,B,C,D]=tf2ss([1],[1 5 10])

可得 $\boldsymbol{A}=\begin{bmatrix}-5 & -10\\ 1 & 0\end{bmatrix}$ $\boldsymbol{B}=\begin{bmatrix}1\\ 0\end{bmatrix}$ $\boldsymbol{C}=[0\ \ 1]$ $\boldsymbol{D}=0$

所以系统的状态方程为

$$\begin{bmatrix}\dot{q}_1\\ \dot{q}_2\end{bmatrix}=\begin{bmatrix}-5 & -10\\ 1 & 0\end{bmatrix}\begin{bmatrix}q_1\\ q_2\end{bmatrix}+\begin{bmatrix}1\\ 0\end{bmatrix}x(t)$$

$$y(t)=[0\ \ 1]\begin{bmatrix}q_1\\ q_2\end{bmatrix}$$

例2　已知某连续系统的状态方程和输出方程为

$$\begin{bmatrix}\dot{q}_1(t)\\ \dot{q}_2(t)\end{bmatrix}=\begin{bmatrix}2 & 3\\ 0 & -1\end{bmatrix}\begin{bmatrix}q_1(t)\\ q_2(t)\end{bmatrix}+\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\end{bmatrix}$$

$$\begin{bmatrix}y_1(t)\\ y_2(t)\end{bmatrix}=\begin{bmatrix}1 & 1\\ 0 & -1\end{bmatrix}\begin{bmatrix}q_1(t)\\ q_2(t)\end{bmatrix}+\begin{bmatrix}1 & 0\\ 1 & 0\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\end{bmatrix}$$

其初始状态和输入分别为

$$\begin{bmatrix}q_1(0^-)\\ q_2(0^-)\end{bmatrix}=\begin{bmatrix}2\\ -1\end{bmatrix}\qquad \begin{bmatrix}x_1(t)\\ x_2(t)\end{bmatrix}=\begin{bmatrix}u(t)\\ \mathrm{e}^{-3t}u(t)\end{bmatrix}$$

求该系统的系统函数矩阵$\boldsymbol{H}(s)$ 和输出。

计算系统函数矩阵$\boldsymbol{H}(s)$

```
A=[2 3;0 –1];B=[0 1; 1 0];
C=[1 1; 0 –1];D=[1 0; 1 0];
[num1,den1]=ss2tf(A,B,C,D,1)
[num2,den2]=ss2tf(A,B,C,D,2)
```

运行结果

```
num1 =1    0   –1
      1   –2    0
den1 =1   –1   –2
num2 =0    1    1
      0    0    0
den2 =1   –1   –2
```

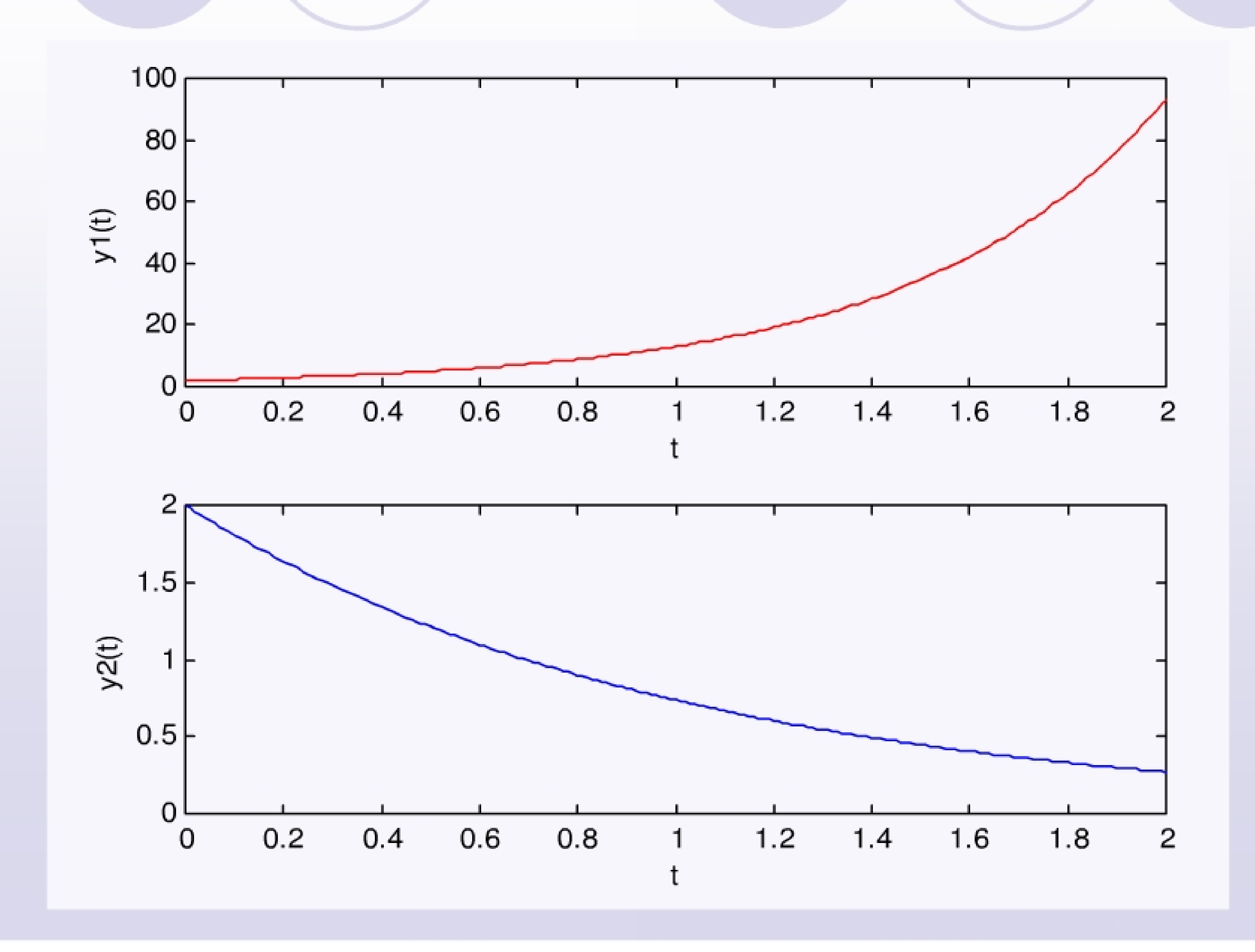
所以系统函数矩阵$\boldsymbol{H}(s)$为

$$\boldsymbol{H}(s)=\frac{1}{s^2-s-2}\begin{bmatrix} s^2-1 & s+1 \\ s^2-2s & 0 \end{bmatrix}$$

计算输出

```
%Program 9_1
A=[2 3;0 -1];B=[0 1; 1 0];
C=[1 1; 0 -1];D=[1 0; 1 0];
q0=[2 -1];
dt=0.01;t=0:dt:2;
x(:,1)=ones(length(t),1);
x(:,2)=exp(-3*t)';
sys=ss(A,B,C,D);
y=lsim(sys,x,t,q0);
subplot(2,1,1);plot(t,y(:,1),'r');
ylabel('y1(t)');xlabel('t');
subplot(2,1,2);plot(t,y(:,2));
ylabel('y2(t)');xlabel('t');
```

运行结果



100
80
60
40
20
0
y1(t)
0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2
t
2
1.5
1
0.5
0
y2(t)
0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2
t

例3　已知某离散系统的状态方程和输出方程为

$$\begin{bmatrix} q_1[k+1] \\ q_2[k+1] \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -2 & 3 \end{bmatrix} \begin{bmatrix} q_1[k] \\ q_2[k] \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} x_1[k] \\ x_2[k] \end{bmatrix}$$

$$\begin{bmatrix} y_1[k] \\ y_2[k] \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} q_1[k] \\ q_2[k] \end{bmatrix}$$

其初始状态和输入分别为

$$\begin{bmatrix} q_1[0] \\ q_2[0] \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \qquad x[k] = u[k]$$

求该系统的输出。

计算输出

```
%Program 9_2
A=[0 1; −2 3];B=[0;1];
C=[1 1; 2  −1];D=zeros(2,1);
q0=[1; −1];
N=10;x=ones(1,N);
y=dlsim(A,B,C,D,x,q0);
subplot(2,1,1);y1=y(:,1)';
stem((0:N−1),y1);
xlabel('k');ylabel('y1');
subplot(2,1,2);y2=y(:,2)';
stem((0:N−1),y2);
xlabel('k');ylabel('y2');
```

运行结果

